تم تحميل الملف من موقع
البوصلة التقنية
www.boosla.com

المدوجز في قواعد البيانات

Fundamentals of Database Systems

المؤلف

جعفر الطيطي

مدرس حاسوب تقني

كلية مجتمع المرأة برام الله

June 23, 2003

الإهداء

إلى أمتي الإسلامية ، علها تدرك أن سبب عزتنا هو الإسلام، وأن الإسلام لن يطبق إلا بوجود دولة الخلافة الراشدة القادمة بإذن الله....
إلى روح والدي الفقيد المربي الفاضل المرحوم إسحق الطيطي... قدوتي و مناري في الحياة...
إلى أمي الشمعة التي لا تنطفئ لتنير لي الدرب....
إلى كل من ساهم في أنجاز هذا الجهد المتواضع...

المقدمة

مع التطور العلمي الذي نشهده في القرن الحادي و العشرين، أخذت المعرفة الإنسانية بالأتساع و التضخم الأفقي و العمودي، وظهرت التطبيقات العلمية التي تحتاج إلى الاحتفاظ بالبيانات المعرفية بشكل سليم من اجل الأجيال القادمة.

فمن مشروع الخارطة الجينية للبشر(GNOME)، إلى مشروع المعلومات الجغرافية(GIS)، إلى الاستعمالات العصرية من مثل التجارة الالكترونية، و أنظمة الاتصالات، وسجلات الهواتف و الموظفين، ونظراً للتضخم الهائل في حجم المعلومات و نوعها أصبح هنالك ضرورة ملحة لعمليات و الفرز و الحفظ و الاسترجاع في المكان و الزمان المناسبين بالدقة والسرعة والتكلفة المناسبة، ظهرت الحاجة إلى استخدام قواعد البيانات و تطبيقاتها.

ومع توفر بعض المراجع العربية في تدريس قواعد البيانات و أساسياتها، إلا إنني وبعد الإطلاع عليها من خلال التدريس بها في كلية مجتمع المرآة بالطيرة، وجدت الكثير من المعلومات القديمة و الناقصة في هذه المراجع، فلذلك أصبحت الحاجة ماسة إلى مادة نظرية مساعدة لهذه المراجع، تعوض النقص في معلوماتها و تحدث القديم فيها.

فلذلك كان هذا الجهد المتواضع، و أرجو الله العلي القدير أن يجعله في ميزان حسناتي، و أن يستفيد منه طلابنا وطالباتنا الأعزاء كل الاستفادة.

م. جعفر الطيطي

جدول المحتويات

الإهداء 1

المقدمة 2

جدول المحتويات 3

الوحدة الأولى: أساسيات قواعد البيانات وتطورها تاريخيا 7

الأهداف العامة للوحدة: 7

ما هي قواعد البيانات ؟ 7
التوجهات السابقة (previous approaches): 7
ما هي قاعدة البيانات ؟ 7
بيئة قواعد البيانات: 8

ما هي أطراف قاعدة البيانات؟ 8
1. المستخدمون (Users): 8
2. البرامج (Software S/W): 8
3. البيانات(Data): 9
4. المعدات (Hardware) 9

خطوات تصميم قواعد البيانات (Design process) 9

أفضلية(Advantages) قواعد البيانات على نظام الملفات القديم 9

فوائد قواعد البيانات(Benefits) 10

مساوئ قواعد البيانات 10

أنواع أنظمة إدارة قواعد البيانات(DBMS) 11
من ناحية المستخدمين: 11
من ناحية التوزيع الجغرافي: 11
من ناحية خطة البيانات(Data Models) 11
1. قواعد البيانات الهرمية (Hierarchical Model) 12
2. قواعد البيانات الشبكية (Network Model) 12
3. قواعد البيانات العلائقية (Relational DB Model): 13

هيكلية قاعدة البيانات (Architecture of a DB system) 14

استقلالية البيانات (DI: Data Independance) 15

أسئلة الوحدة: 16

الوحدة الثانية: التخزين الفيزيائي لقواعد البيانات(DB STORAGE) 17

الأهداف العامة للوحدة: 17

واسطات التخزين (Storage Devices) 17

عملية الحصول على سجل معين 19

تنظيم الملفات(File organization) 21
أنواع السجلات: 21
1. من ناحية الطول(Length): 21
2. من ناحية التوزيع(Spanning): 22

طرق تنظيم الملفات 22
1. الملفات غير المرتبة (Heap File Unordered) 22
2. الملفات المرتبة (Ordered Files) 23
3. باستخدام طريقة المزج(Hashing Techniques): 24
1. المزج الداخلي(Internal Hashing): 24
2. المزج الخارجي(External Hashing): 25

الفهارس وتراكيب الوصول (Indexes and Access Structure) 26

الفهارس الثابتة المفردة(Static single Index): 27
1. الفهارس الرئيسية(Primary Index). 27
2. الفهارس الثانوية(Secondary Index). 28
1. بدون تكرار(Not-repeated): 28
2. وجود تكرار(repeated): 29
الأفضليات و السيئات للفهرس 29

أسئلة الوحدة: 30

الوحدة الثالثة: تصميم قواعد البيانات العلائقية (RELATIONAL DATABASE DESIGN) 31

الأهداف العامة للوحدة: 31

خطوات تصميم قواعد البيانات(Database Design Steps) 31

نموذج العلاقات و الكينونات (E-R Model) 32
1. الكيانات(Entities). 32
2. نوعية كيان(Entity Type): 32
3. مجموعة كيان(Entity Set): العناصر 33
5. أنواع الخصائص(Types of Attributes) 33
6. مجال الخاصية(Domain of Attribute) 34
7. العلاقات المشتركة(Relationships) 34
8. الدرجة للعلاقة المشتركة(Relationship Degree): 34
رسم نموذج E-R Model: 34
الضوابط البنائية (Structural Constraints) 37
نوعية الكيانات الضعيفة(WE: Weak Entity type) 37

مثال كامل على نموذج العلاقات الكينونات(قاعدة البيانات لجامعة) 39
المطلوب: 40
خطوات الحل: 40

النموذج العلائقي(Relational Model) 42

1. العلاقة (Relation): 42
2. المجال(Domain): 43
3. درجة العلاقة(Relation Degree): 43
4. جوهرية العلاقة (Relation Cardinality): 44
5. قاعدة البيانات العلائقية(Relational DB) 44

ضوابط قواعد البيانات العلائقية (Relational Db Constraints) 44
1. ضوابط المجال(Domain Constraints) 44
2. ضوابط المفاتيح(Key Constraints): 44
أ- المفاتيح الرئيسية (PK: Primary Key). 44
ب- المفتاح الغريب(Foreign Key): 45
3. القواعد الضبطية (Constraints Rules): 46
أ- صحة الكيان (Entity Integrity): 46
ب- صحة التوزيع (Referential Integrity): 46

تحويل نموذج العلاقات و الكينونات إلى النموذج العلائقي (ER To Relational) 46

أسئلة الوحدة: 49

الوحدة الرابعة: التطبيع(NORMALIZATION) 51

الأهداف العامة للوحدة: 51

الإعتمادات الوظيفية (FD: Functional Dependencies) 51

نماذج التطبيع 52
- **تعريف التطبيع: 52**

نماذج التطبيع الموجودة: 53
(1) نموذج التطبيع الأول (1NF): 53
(2) نموذج التطبيع الثاني(2NF): 54
(3) نموذج التطبيع الثالث(3NF): 55
(4) نموذج تطبيع بويس كود(BCNF: Boyce-Codd Normal Form): 55
(5) نموذج التطبيع الثاني(2NF) حسب بويس كود: 56
(6) نموذج التطبيع الثالث(3NF) حسب بويس كود: 56
(7) نموذج بويس كود المعدل: 56

أسئلة الوحدة: 57

الوحدة الخامسة: لغة الاستعلام البنيوية (SQL) 58

الأهداف العامة للوحدة: 58

مقدمة: 58

قصة SQL: 59

ما هي لغة SQL ؟ 60
العلاقة بين SQL و بين الجبر العلائقي 60
جمل تعريف البيانات(DDL) و جمل معالجة البيانات(DML) في لغة SQL 61

قاعدة البيانات لشركة(Company Database): 62

جملة الاختيار (SELECT) **63**
أقسام جملة SELECT الرئيسية **63**
1. قسم الاختيار(Select clause) 64
2. قسم الشرط(Where clause) 65
3. قسم مصدر البيانات(from clause) 65
معالجة السلاسل الرمزية (String Operations) **67**
ترتيب النتائج (Ordering Results) **68**
العمليات الحسابية (Aggregate Functions) **68**

جمل التغيير على البيانات (Date changing) **70**
إضافة سجلات(Insert Into) **70**
حذف سجلات(delete) **70**
تعديل السجلات(Update) **71**

جمل تعريف البيانات (DDL) **71**
جملة (Create) **71**
جملة انشأ الجداول (Create Table) **72**
جملة حذف الجداول (Drop Table) 72
جملة تغيير التصميم لجداول (Alter Table) **73**
جملة انشأ فهرس بيانات (Create Index) **73**
جملة انشأ مظهر بيانات (Create View) **73**

أسئلة الوحدة: **74**

المراجع **75**

ملحق: قاعدة بيانات لشركة **76**

الوحدة الأولى: أساسيات قواعد البيانات وتطورها تاريخيا

الأهداف العامة للوحدة:

أن يتعرف الطالب/ة على:

1. الطرق السابقة في تنظيم البيانات.
2. تعريف قواعد البيانات و بيئتها.
3. أطراف قواعد البيانات.
4. أفضليات وفوائد قواعد البيانات.
5. أنواع ألDBMS.
6. نماذج أنظمة قواعد البيانات:
   i. قواعد البيانات الهرمية.
   ii. قواعد البيانات الشبكية.
   iii. قواعد البيانات العلائقية.
7. مقارنة بين النماذج المختلفة لأنظمة قواعد البيانات
8. هيكلية قواعد البيانات العلائقية.
9. استقلالية البيانات.
10. المناظر(Views).

ما هي قواعد البيانات ؟

التوجهات السابقة (previous approaches):

1. قواعد البيانات اليدوية (manual DB): مثل بطاقات المكتبة لذلك يوجد مشاكل.

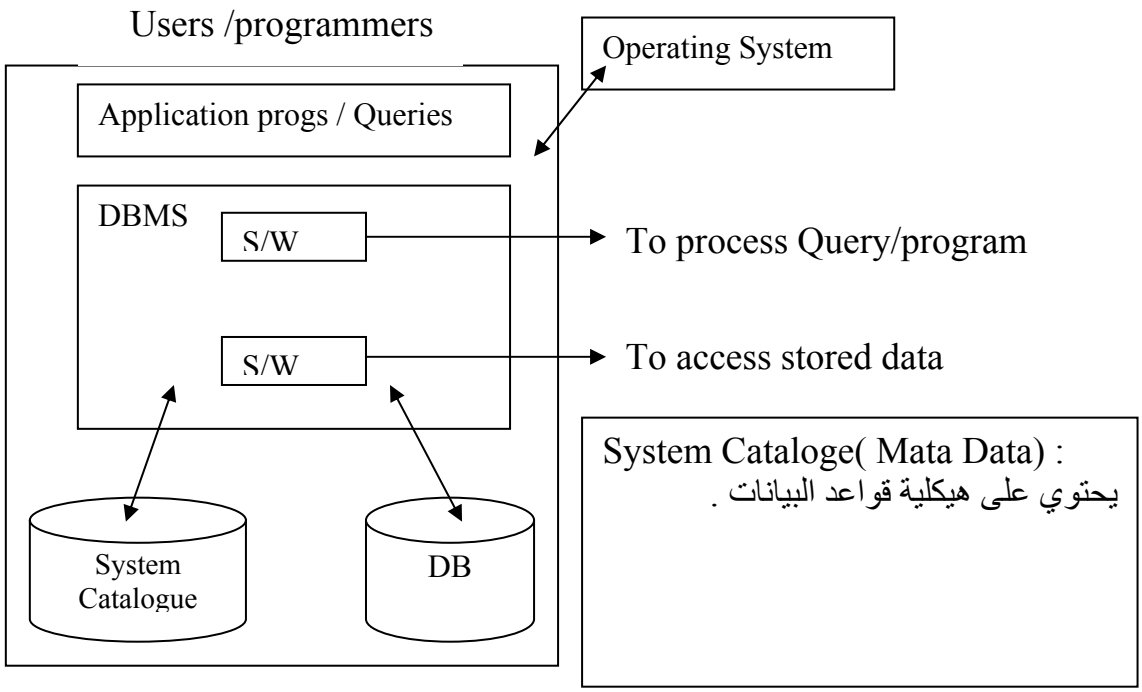
الموضوع	المؤلف	العنوان
Data	Data	Data ....

2. أنظمة الملفات القديمة (traditional filling System).
3. قواعد البيانات المحوسبة (computerized DB), وهي المستخدمة في الوقت الحاضر.

ما هي قاعدة البيانات ؟

قاعدة البيانات هي عبارة عن مجموعة من المعلومات (البيانات) المترابطة, والتي تسجل ولها معنى مفهوم. أنها تصمم وتبنى وتعبأ بالبيانات لهدف معين, ولها مستخدموها وبرامجها التطبيقية.

بيئة قواعد البيانات:



ما هي أطراف قاعدة البيانات؟

1. المستخدمون (Users):

أ- المستخدمون الطرفيون (end users):
أستعلام وتحديث (إدخال, حذف, تغيير) قاعدة البيانات.

ب- المبرمجون للبرامج التطبيقية (application programs):
واجهات التطبيق, الاستعلامات, والبرمجة.

ج- مصمم قاعدة البيانات(DA or db designer):
تحليل النظم و دراستها.

د- مشرف قواعد البيانات (Database Administrator ((DBA)):
يعطي الإمكانيات المتاحة لكل مستخدم وتشمل الحماية, البرامج (Software), والمعدات (hardware).

2. البرامج (Software S/W):

1. البرامج التطبيقية.

2. برنامج إدارة قواعد البيانات (Database Management System)(DBMS):
وهو عبارة عن مجموعة من البرامج التي تمكن المستخدمين من أنشاء والمحافظة على قاعدة البيانات:-
أ- تعريف قاعدة البيانات (التركيب, أنواع البيانات, والشرح).
ب- بناء قاعدة البيانات (الحفاظ على التعريف الثانوي).
ج- التعامل مع قاعدة البيانات(الإستعلامات, التقارير).

3. البيانات(Data):

مجموعه من الحقائق المرتبطة مع بعضها البعض بطريقه مرتبة, وتكون إما خاصة أو عامه (أكثر من مستخدم).

4. المعدات (Hardware)

أ.التخزين الثانوي
ب.التعبير الفيزيائي عن المعلومات

DBM: database machine

خطوات تصميم قواعد البيانات (Design process)

1. **مصمم قاعدة البيانات [Data administrator not DBA]**
يقوم بجمع الحقائق حول المشكلة.

2. **تعريف قاعدة البيانات:**
أ. تعريف الجداول.
ب. تعريف الحقول لكل جدول.
ت. تعريف أنواع البيانات (Data types) لكل حقل.
ث. الضوابط على الحقول (Field constraints).
ج. الضوابط على العلاقات(relation constraints) للمحافظة على صحة المعلومات (Data Integrity).

3. **بناء قاعدة البيانات :**
إدخال المعلومات.

4. **التعامل مع قاعدة البيانات(Manipulation)**
استعلامات و تحديث(update).

أفضلية(Advantages) قواعد البيانات على نظام الملفات القديم

1. طبيعة الوصف الذاتي لقاعدة البيانات(system catalogue).

2. استقلالية البيانات (Data Independence):
   1. فيزيائيا(Physically):
      البرامج التطبيقية لا تعتمد على التعريف الفيزيائي لقاعدة البيانات.
   2. منطقيا (Logically) :
      تصميم قاعدة البيانات (DB) مستقل عن نوع نظام إدارة قواعد البيانات (DBMS)المستخدم. ولكن في نظام الملفات القديم يجب أن يعرف كل نوع بيانات مستخدم ونقطة البداية والنهاية له.
   3. تدعم عدة مناظر للبيانات (Multiple views).
   4. تدعم الأنظمة متعددة المستخدمين.

فوائد قواعد البيانات(Benefits)

1. السرعة, الدقة, حفظ المساحة (save coding).
2. التحكم بمقدار المعلومات التي لا حاجه لها:
   by DBMS(Controlling Redundancy).
3. التخلص من عدم التوافق (Inconsistency)باستخدام تعميم التغييرات على كل العلاقات.
4. فرض ضوابط صحة(تكامل) المعلومات(Enforcing Integrity Constraints)
   وتشمل هذه الضوابط:
   أ. الضوابط على البيانات.
   ب. التكامل الضبطي (referential integrity) :
      باستخدام المفاتيح (PK,FK) : حيث
      PK: Primary Key (المفتاح الرئيسي).
      FK: Foreign Key (المفتاح الغريب).
   ج. التفرد في القيمة (uniqueness) : مثل المفتاح الرئيسي, رقم الطالب وحيد لكل طالب.
5. فرض المواصفات و المقاييس العلمية (standards): مثل إنشاء التقارير عن البيانات بشكل قياسي.
6. الحفظ للنسخة الإضافية و الاستعادة في حال حدوث مشاكل (backup and recovery).
7. الحماية و تحديد الإمكانيات لكل مستخدم, عن طريق اسم المستخدم وكلمة السر.

مساوئ قواعد البيانات

مع أن فوائد قواعد البيانات جمة, إلا أن لها بعض المساوئ الجانبية:
1. التكاليف الكبيرة على المعدات و البرامج و البشر.
2. تحتاج إلى معدات قوية.

أنواع أنظمة إدارة قواعد البيانات(DBMS)

من ناحية المستخدمين:

أ. مستخدم واحد(Single User).
ب. متعدد المستخدمين (Multi-User).

من ناحية التوزيع الجغرافي:

أ. أنظمة إدارة قواعد البيانات المركزية (Centralized DBMS).
ب. أنظمة إدارة قواعد البيانات الموزعة (Distributed DBMS).

من ناحية خطة البيانات(Data Models)

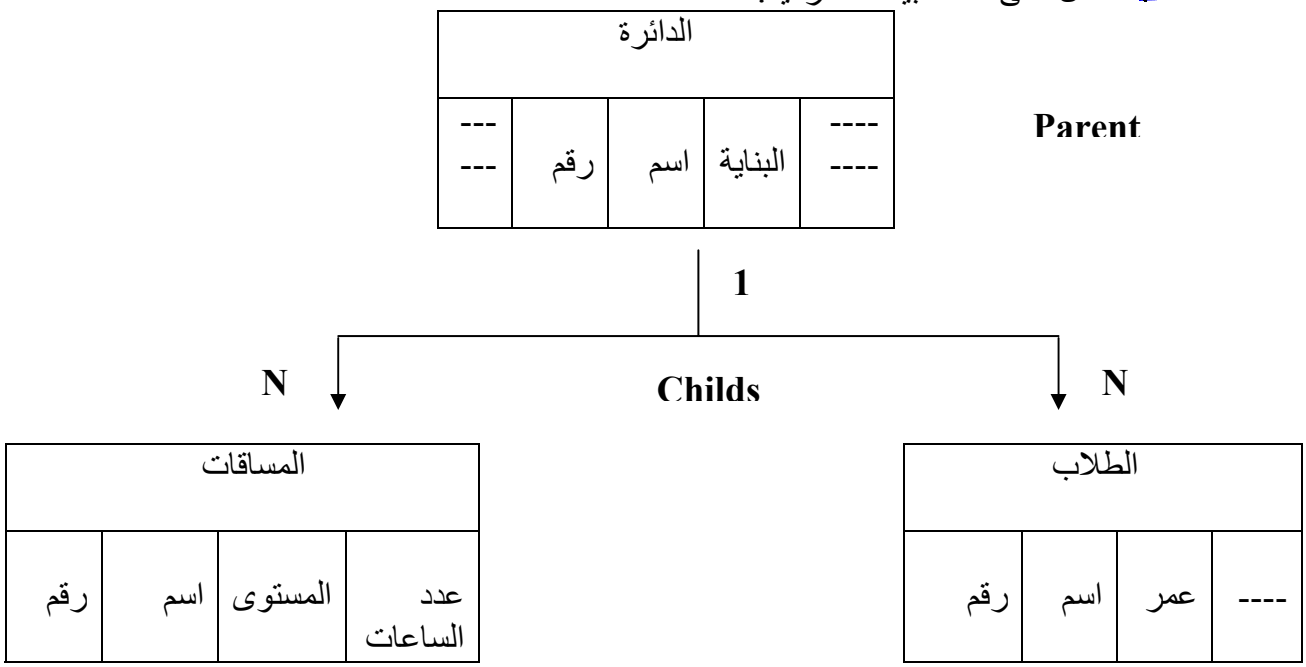
تعرف خطة البيانات(Data Model or schema) بأنها مجموعة من المفاهيم التي تستخدم في وصف تركيب قاعدة البيانات.
وهي أيضا عبارة عن تعريف لعمليات المستخدم الممكنة على قاعدة البيانات.

فمن ناحية خطة البيانات يوجد هنالك ثلاثة أنواع من أنظمة إدارة قواعد البيانات:
1. أنظمة إدارة قواعد البيانات الهرمية (Hierarchical DBMS).
2. أنظمة إدارة قواعد البيانات الشبكية (Network DBMS).
3. أنظمة إدارة قواعد البيانات العلائقية (Relational DBMS).

1. قواعد البيانات الهرمية (Hierarchical Model)

- أقدم أنواع أنظمة قواعد البيانات،وفي هذه الخطة ترتبط البيانات مثل الفروع للشجرة مع الشجرة نفسها ،بحيث أن السجلات ترتبط بطريقة علاقة الوالد للولد.

- Parent and child relationships (1→N)
- مثال: الأب له عدة أولاد
- مثال على قاعدة بيانات هرمية:



- من أهم الشركات التي عملت على قواعد البيانات الهرمية، هي شركة IBM وكان اسم المنتج IMS في عام 1960 وكانت تستخدم لغة COBOL في عمليات الاستعلام والتجديد.

2. قواعد البيانات الشبكية (Network Model)

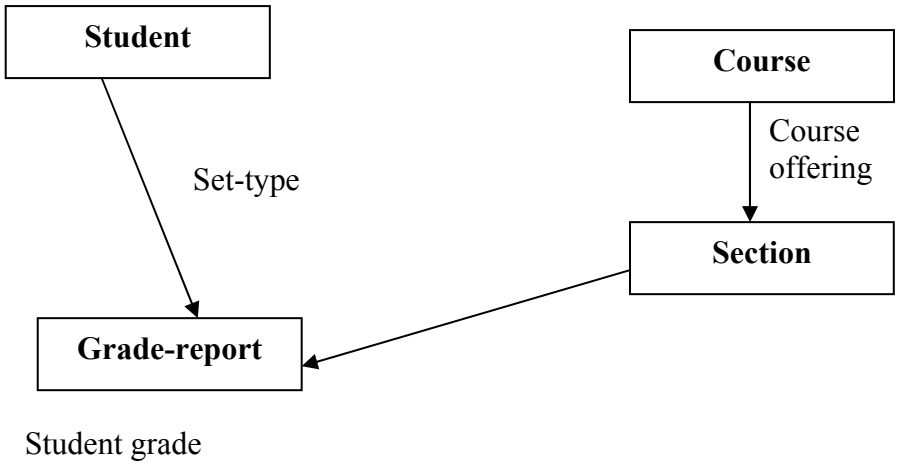
- طورت على أساس قواعد البيانات الهرمية، لتمثيل المعلومات التي له خاصية لا تسمح بتمثيلها بعلاقة الأب والابن في قواعد البيانات الهرمية.
- تمثل السجلات بشكل نوعية سجلات والعلاقات بشكل مجموعة.
- أنواع العلاقات المسموح بها:

1→1, 1→N, M→N

- طورت بواسطة مؤتمر:

CODASYL (Common Data System Language)

- مثال على قاعدة بيانات شبكية:

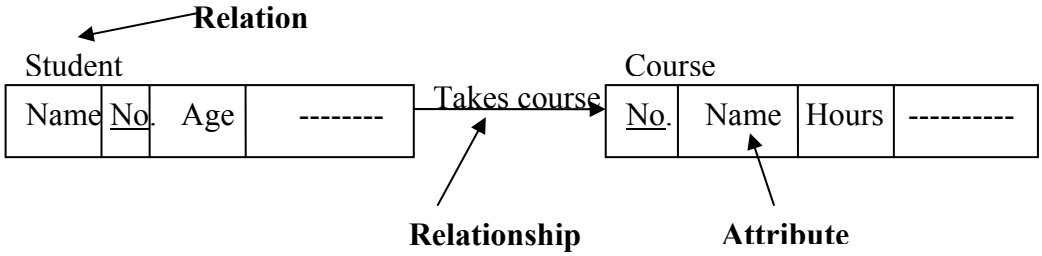


- من أهم الشركات و منتجاتها: IBM (IDMS/R), DEC (DBMS)

ولكن هذان النوعان من أنظمة قواعد البيانات لا تستخدم في هذه الأيام وإنما المسيطر على سوق قواعد البيانات هو قواعد البيانات العلائقية.

3. قواعد البيانات العلائقية (Relational DB Model):

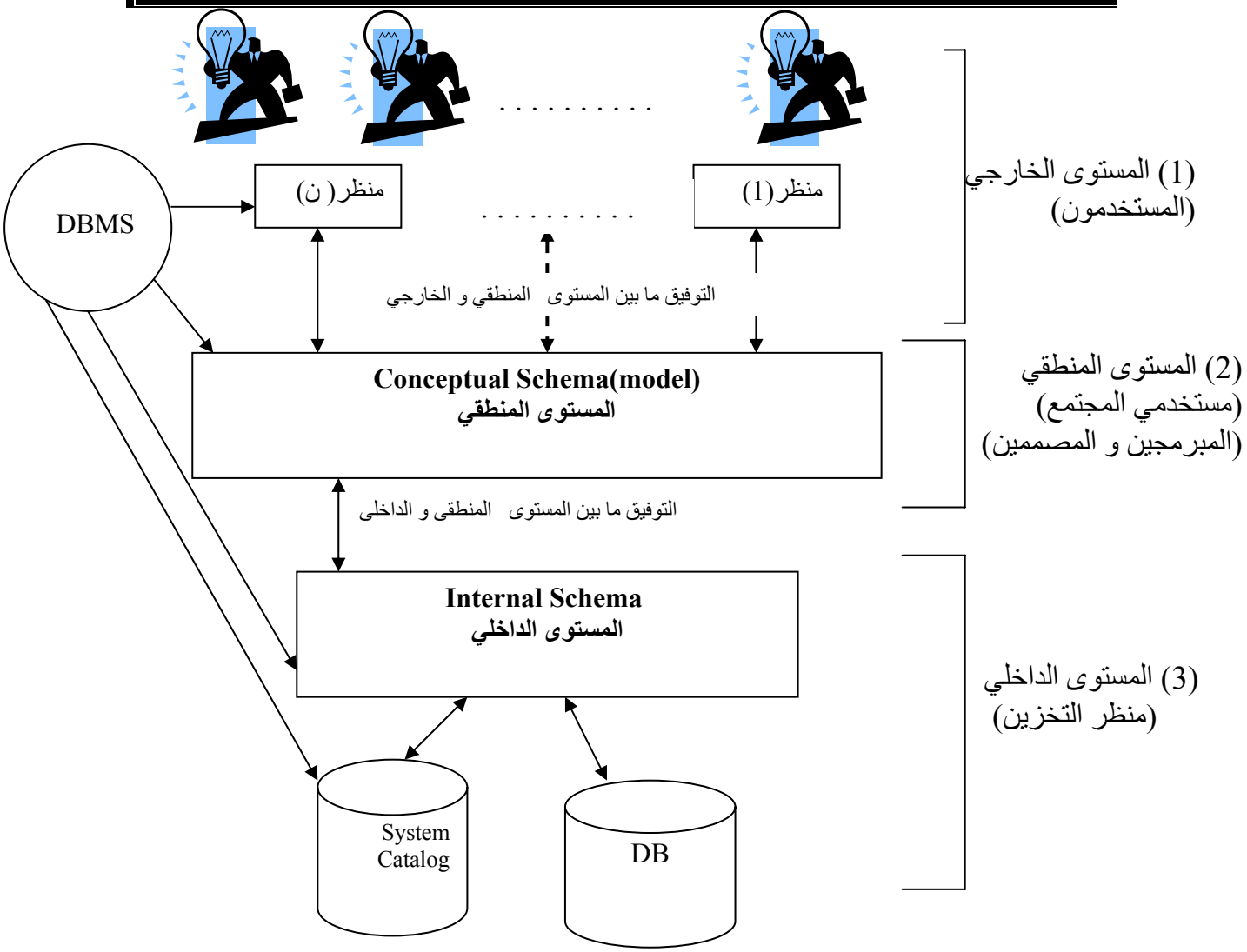
- أخترعها Codd عام 1971.
- لغة SQL, PL-SQL هي لغة الاستعلام في هذه الخطة للبيانات.



- هي المستخدمة في الأسواق حاليا ومن الأمثلة عليها:

Access, Oracle, FoxPro, SQL SERVER

هيكلية قاعدة البيانات (Architecture of a DB system)



- اللغه المستخدمه كما قلنا هي SQL (Structured Query Language)
وتقسم إلى نوعين :-
1. DDL (Data Definition Language) : لغة تعريف البيانات.
2. DML (Data manipulation Language) : لغة التعامل مع البيانات.

- امثلة على المستويات الثلاث
1. المستوى الخارجي:
بإستخدام SQL(DML):

```
Create view saddress
As
Select SNo ,address
From student
```

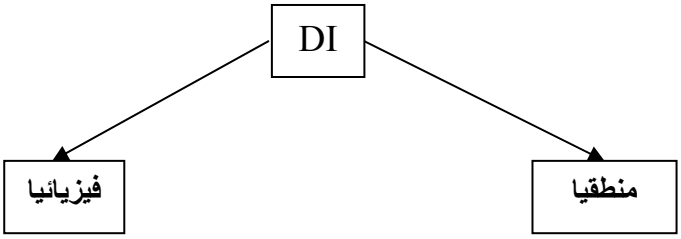
2. المستوى المنطقي
باستخدام SQL (DDL):

```
Create table student
SNo char(8),
Sname char(20),
Dept#,
Primary key SNo,
Foreign key Dept# refers dept table;
```

3. المستوى الداخلي
التخزين الداخلي

```
Stored student rec 40 Byte
SNo Type = Byte (8), offset (6);
...
…
```

استقلالية البيانات (DI: Data Independance)



التعريفات

استقلالية البيانات:
القدرة على التغير في خطة احد المستويات بدون التأثير على المستويات التي أعلى منها.

استقلالية البيانات منطقياً(Logical DI)
القدرة على التغير في الخطة المستوى المنطقي بدون الحاجة للقيام بأية تغيرات أخرى للمستوى الخارجي

استقلالية البيانات فيزيائياً (physical DI)
القدرة على التغير في خطة المستوى الداخلي بدون الحاجة للقيام بأية تغيرات أخرى على لمستوى المنطقي أو الخارجي.

أسئلة الوحدة:

س[1]) بين ما هو التعريف لكل من المفاهيم التالية :
1) قاعدة البيانات.
2) مشرف قاعدة البيانات.
3) استقلالية البيانات المنطقية.

س[2]) بين ما المقصود بخطة البيانات(Data Model) و اشرحي أنواع أنظمة إدارة قواعد البيانات من ناحية خطة البيانات مع ذكر ميزات و سيئات كل نظام؟

س[3])ادرس احد الشركات التي تحتوي كل الدوائر و الموظفين ومشاريع وبين فيه المستويات لتمثيل البيانات الثلاث (الخارجية والمنطقية والداخلية).

الوحدة الثانية: التخزين الفيزيائي لقواعد البيانات(DB (Storage

الأهداف العامة للوحدة:

أن يتعرف الطالب/ة على:

1. واسطات التخزين (Storage Devices) و عملية الحصول على سجل معين منها.
2. تنظيم الملفات(File organization) وأنواع السجلات: من ناحية الطول والتوزيع.
3. طرق تنظيم الملفات: غير المرتبة و المرتبة و باستخدام طريقة المزج.
4. الفهارس وتراكيب الوصول وأنواعها.
5. الأفضليات و السيئات للفهرس.

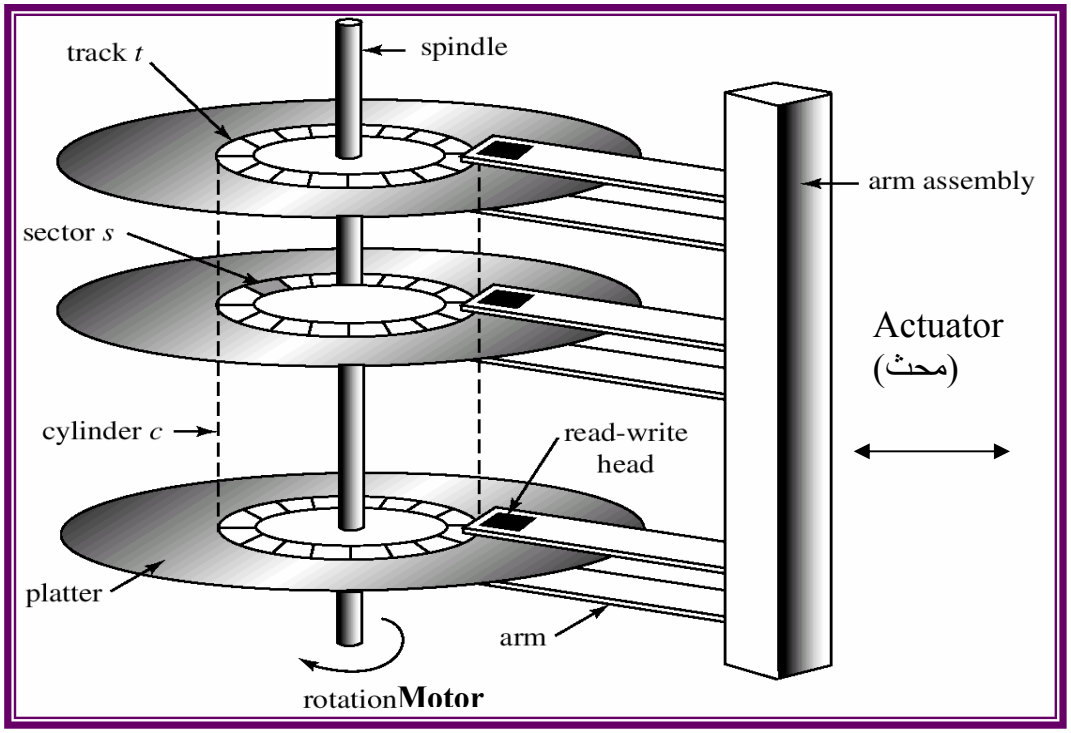
واسطات التخزين (Storage Devices)

- يوجد نوعين من التخزين:

أ. التخزين الرئيسي(primary storage)
يمتاز بأنه مؤقت(volatile)، يعني عند انطفاء الجهاز تضيع البيانات
ب. التخزين الثانوي(Secondary Storage)

1. Disks (الأقراص) → وصول مباشر (Direct access).
2. Tapes (الأشرطة) → وصول تتابعي (Sequential access).

- الشكل المرسوم يمثل شكل القرص الصلب وكيفية عمله:



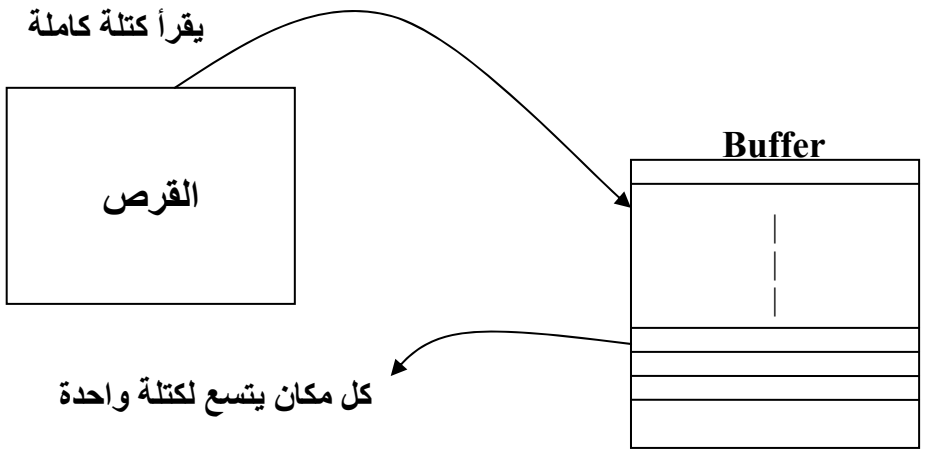
- ولكن سرعة الوصول على الأقراص ليست وصولا عشوائيا فعليا بسبب وجود بعض أوقات التأخير التالية:
  أ. وقت البحث (seek time): وهو الوقت الذي يحتاجه المحث(Actuator) للوصول إلى الأسطوانة(Cylinder) المطلوبة.
  ب. وقت الدوران(Rotation Time): وهو الوقت الذي يحتاجه القرص حتى يلف إلى القطاع المطلوب.
  ج. وقت النقل(Transfer Time): وقت النقل للبيانات إلى الذاكرة الرئيسية.

- مجموع هذه الأوقات يسمى وقت الوصول(Access Time).

- ولذلك يسمى الوصول إلى القرص الصلب وصولاً مباشراً(Direct Access)، وليس و صولاً عشوائياً كما هو في الذاكرة الرئيسية للحاسوب.

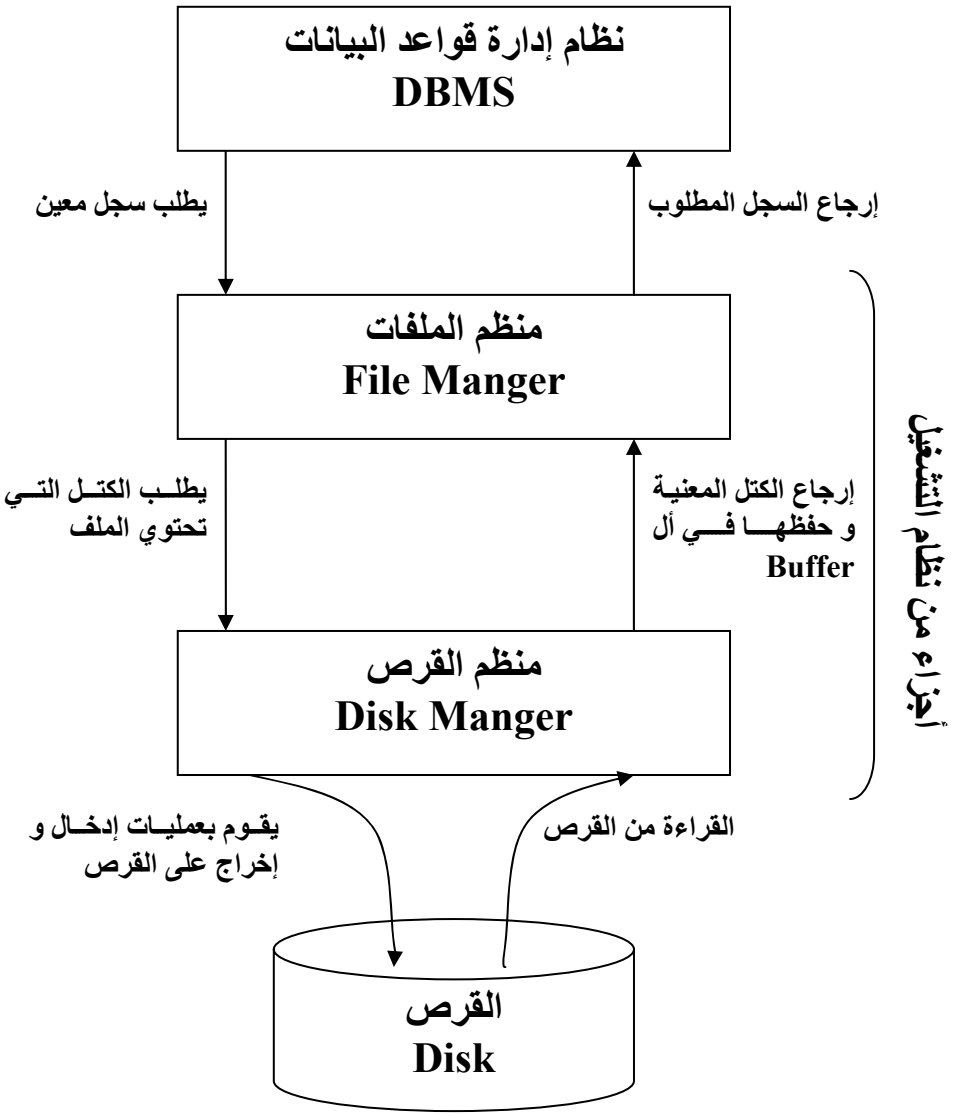
ملاحظة:

يستطيع رأس القراءة وفي المرة الواحدة قراءة كتلة بيانات كاملة (Block or sector)، ويضعها في مكان في الذاكرة الرئيسية يسمى Buffer، حيث أن كل و حدة سعة داخل أل Buffer تتسع لكتلة واحدة كما هو مبين في الشكل التالي:



- مع العلم أن حجم الكتلة ثابت للقرص الواحد.

عملية الحصول على سجل معين



- لا ننسى بان الملف هو عبارة عن مجموعة من السجلات و أيضا الملف قد يكون محفوظاً على كتلة أو مجموعة من الكتل.

مثال:

لنفرض أن عندنا قاعدة البيانات لشركة استيراد تحتوي على ثلاثة جداول, و هذه الجداول هي جدول المورد (Supplier)، جدول القطع (Parts)، وجدول المورد-القطع (SP). كما هو موضح في الشكل التالي:

(1) Supplier (S):

S#	Sname	address
S1	ABC	….
S2	ABE	…
S3	.	
.	.	
.	.	
.	.	

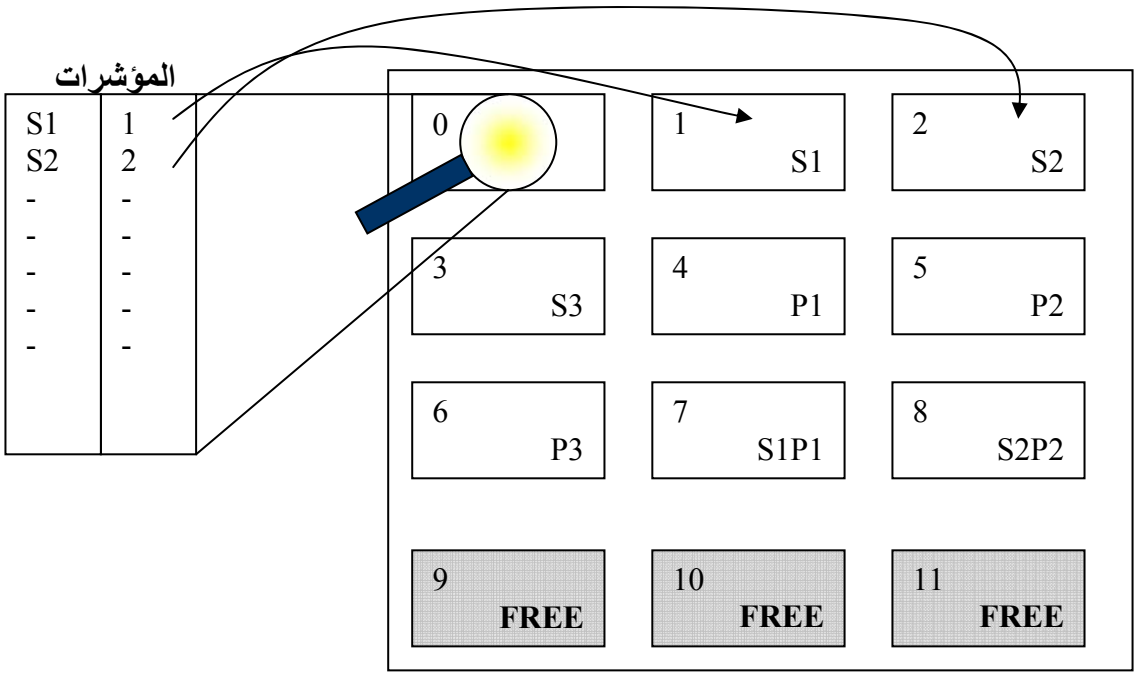
(2) Parts (P):

P#	Partnam	color
P1	screen	….
P2	pins	…
P33	CD	
.	.	
.	.	
.	.	

(3) SP:

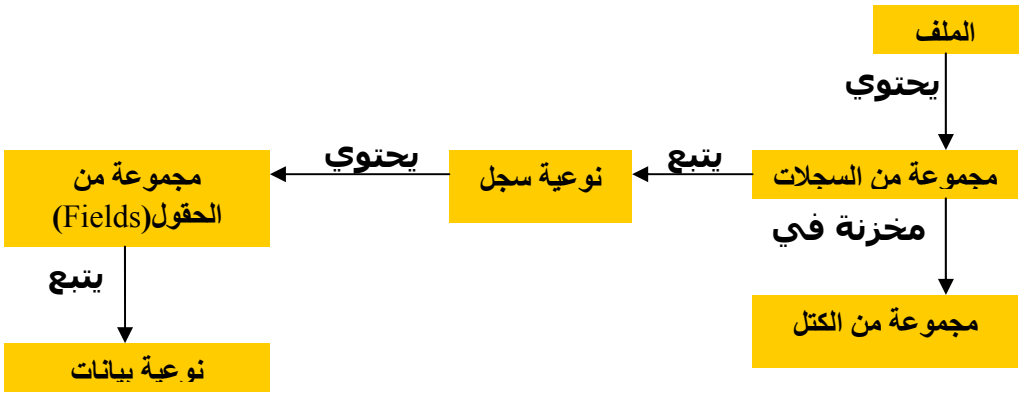
S#	P#	QTY
S1	P1	200
S1	P2	300
S2	P1	400
S3	P3	500

و يمكن تمثيل التخزين على القرص كالتالي:



ولكن بعد فترة من الحذف والإضافة تصبح البيانات مبعثرة, ولذلك نستخدم المؤشرات(Pointers) من أجل الحصول على الوصول المباشر.

تنظيم الملفات(File organization)



أنواع السجلات:

1. **من ناحية الطول(Length):**
   أ. ثابتة الطول.
   ب. متغيرة الطول.

2. **من ناحية التوزيع(Spanning):**
   أ. سجلات قابلة للتوزيع (Spanned records).
   ب. سجلات غير قابلة للتوزيع (Unspanned records).

تعريف:

معامل التكتل (blocking factor):

معامل التكتل(BF) = $\frac{B}{R}$ = $\frac{\text{مساحة الكتلة بالبايت}}{\text{مساحة السجل بالبايت}}$

مثال:

لنأخذ مثالاً على سجلات ثابتة الطول، ومساحة السجل الواحد هي 100 بايت، ولنأخذ مساحة الكتلة الواحدة تساوي 1 كيلوبايت = 1024 بايت، و لنأخذ الحالتين التاليتين:

(1)في حالة أن تكون السجلات قابلة للتوزيع:

$$BF = \frac{1024}{100} = 10.24\,\mathrm{Record / Block}$$

(2)في حالة أن تكون السجلات غير قابلة للتوزيع:

$$BF = \left\lfloor \frac{1024}{100} \right\rfloor = 10\,\mathrm{Record / Block}$$

نعني بتنظيم الملفات:
الطريقة التي بواسطتها تنظم السجلات في التخزين، و هي تعتمد على طريقة الوصول في البحث و الإدخال و التغيير و الحذف.

طرق تنظيم الملفات

1. الملفات غير المرتبة (Heap File Unordered)

تخزن السجلات بنفس الترتيب الذي تدخل به، ويسم هذا النوع من التنظيم:
FCFS: First Come First Served.

مثال:

لو أدخلنا السجلات بالأرقام التالية(تمثل المفتاح الرئيسي PK للسجلات):

100, 200, 310, 400, 500, 300, 600, 201, 320, 210

سترتب في الكتل كم هو موضح في الشكل التالي:

b1	100
	200
	310
	400
b2	500
	300
	600
	201
b3	320
	210

الأفضليات:

(1) إدخال بسيط.

السيئات:

(1) الاسترجاع (البحث من النوع التتابعي الخطي) يأخذ وقت طويل.

(2) في حالة الحذف ستظهر مساحات فارغة غير مستغلة.

ونحل مشكلة الحذف بإضافة بت إضافية تمثل عملية الحذف كالتالي:

1= محذوف 0 = غير محذوف

وبعد الانتهاء من العمل نقوم بإعادة ترتيب السجلات.

2. الملفات المرتبة (Ordered Files)

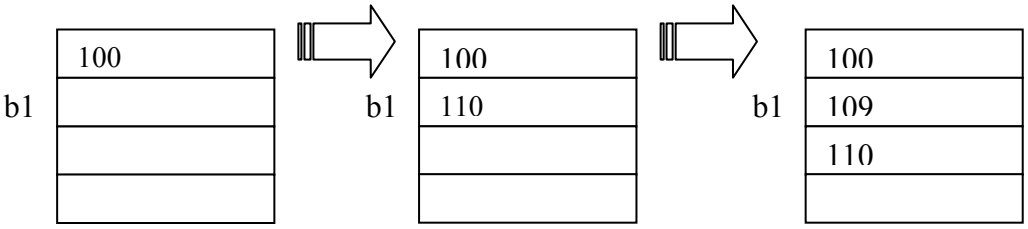
تخزن السجلات بحسب ارقامها، أي يوجد بها ترتيب بحسب البيانات.

مثال:

لو أدخلنا السجلات بالأرقام التالية(تمثل المفتاح الرئيسي PK للسجلات):

100, 110,109

سترتب في الكتل كم هو موضح في الشكل التالي:



الأفضليات:

(1) البحث له فاعلية عالية، بحيث نستطيع استخدام البحث الثنائي على المفتاح الرئيسي, وبسرعة:

$Log_2 (N)$: بحيث أن N هي عدد السجلات.

السيئات:

(1) الإدخال و الحذف يأخذان وقتاً طويلاً، بحيث نحتاج إلى تغيير المواقع(Shift) من أجل الترتيب.

(2) البحث باستخدام مفتاح غير الرئيسي، تكون السجلات غير مرتبة بالنسبة له.

ونحل مشكلة الإدخال والحذف باستخدام فكرة الكتل الانتقالية(Transaction Blocks). وفي أخر اليوم يتم إدخالها أو حذفها من كتل التخزين الحقيقية.

3. باستخدام طريقة المزج(Hashing Techniques):

و يوجد هنالك نوعين من طرق المزج المستخدمة في قواعد البيانات، هي المزج الداخلي و المزج الخارجي:

1. المزج الداخلي(Internal Hashing):

ونقصد به الطريقة التي تدخل بها السجلات على الكتلة باستخدام أقتران المزج التالي:

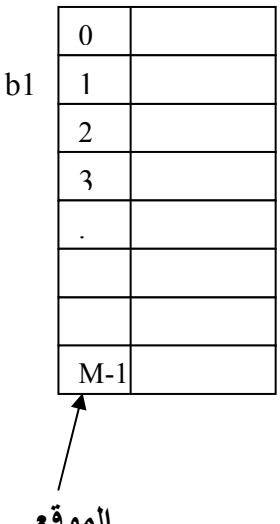
$$h(k) = k \bmod M + 1$$

بحيث إن:

k: قيمة المفتاح الرئيسي.

mod: باقي القسمة.

M: عدد السجلات التي تستطيع الكتلة تخزينها.



الأفضليات:

(1) الوصول المباشر للبيانات (Direct Access).

السيئات:

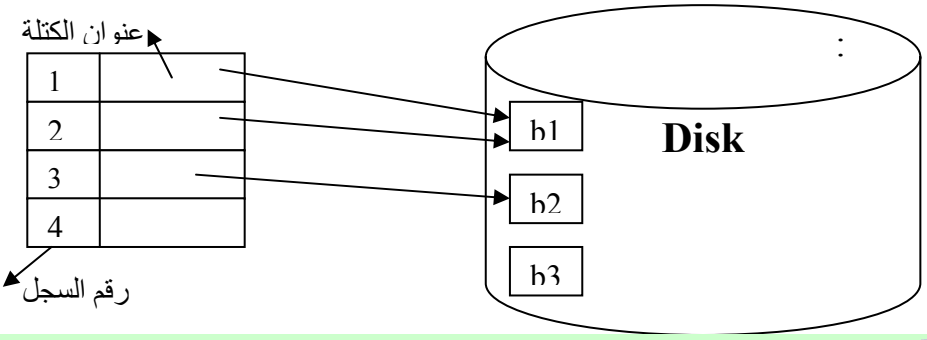
(1) التضارب(collisions)، ونقصد بها أن يأتي سجلان في نفس الموقع.

(2) غير مرتبة للمفاتيح غير المفتاح الرئيسي.

(3) إيضاً نحتاج لحجز المساحة للملف قبل التخزين وذلك يسبب ضياع(Waste) في المساحة الموجودة على القرص.

2. المزج الخارجي(External Hashing):

ونقصد به الطريقة التي يتم بواسطتها توزيع السجلات على الكتل، كما هو موضح في الرسم، ولذلك هنالك وصول مباشر:



مثال:

لنفرض أن لدينا السجلات التالية و نريد أن ندخلها في الكتل (بحيث أن الكتلة الواحدة تتسع ل 9 سجلات):

25, 27, 37, 40, 113, 17.

وكان عندنا أقترانا المزج التاليين:

1. المستوى الداخلي:

$$h_1(k) = k \bmod 9 + 1$$

2. المستوى الخارجي:

$$h_2(k) = k \bmod 3 + 1$$

	b1	b2	b3
1	**27**		
2		**37**	
3			
4			
5		**40**	
6			**113**
7			
8		**25**	
9			**17**

فأن السجلات ستدخل إلى الكتل بالشكل الموضح إلى اليسار:

مثال:

أوجد أقترانات المزج الداخلية و الخارجية لقاعدة البيانات التي تحتوي على جدول(ملف)، بحيث أن هذا الملف يحتوي على 4000 سجل، و حجم السجل الواحدة هي

100 بايت، و الكتلة الواحدة في القرص الصلب تتسع ل 4 كيلوبايت، مع العلم بأن السجلات من النوع غير القابل للتوزيع.

الحل:

$$BF = \left\lfloor \frac{4 \times 1024}{100} \right\rfloor = 40\,\mathrm{Re}\,cord / Block$$

$$Number\ of\ Blocks\ \ needed = \frac{4000\,\mathrm{Re}\,cord}{40\,\mathrm{Re}\,cord / Block} = 100\,Block$$

1. المستوى الداخلي: $h_1(k) = k \bmod 40 + 1$
2. المستوى الخارجي: $h_2(k) = k \bmod 100 + 1$

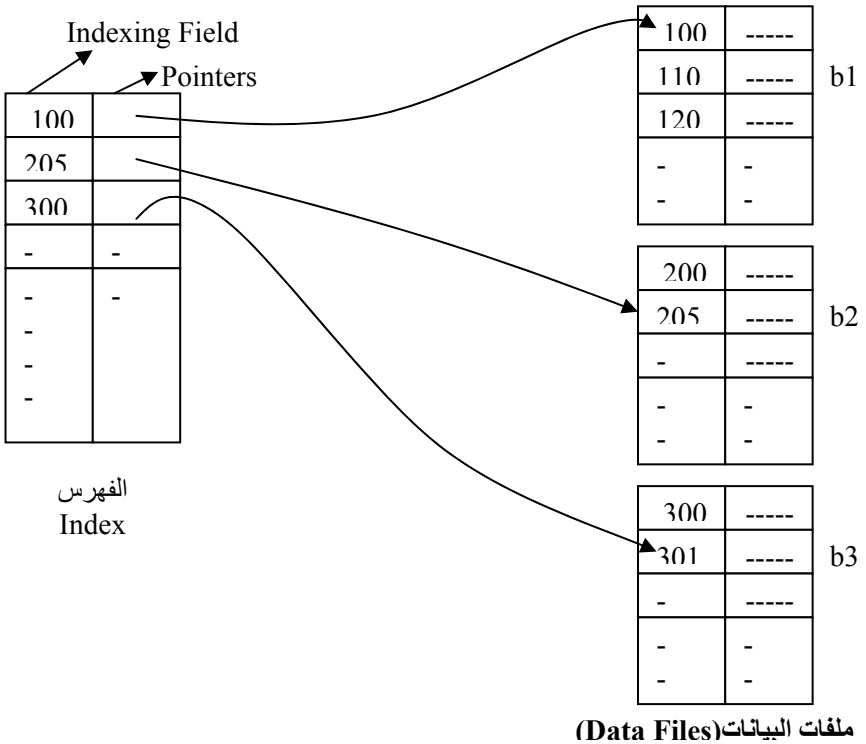
الفهارس وتراكيب الوصول (Indexes and Access Structure)

- الفهارس عادة تستخدم لإسراع البحث في ملف بيانات بالاعتماد على قيم معينة (مفاتيح).

تعريف:

الفهرس:

هو عبارة عن تركيب بيانات، يعرف على أساس احد الحقول (أو مجموعة من الحقول) تسمى حقل الفهرسة(Indexing Filed) وحقل أخر يسمى المؤشر(Pointer)، حيث يؤشر على كتلة معينة أو سجل معين.



يوجد نوعين من الفهارس :

1. فهارس ثابتة(Static Index):

وهي نوعان Single, multiple.

2. فهارس متغيرة(Dynamic Index):

وفي هذا المساق سندرس الفهارس الثابتة فقط من النوع المفردة (single).

الفهارس الثابتة المفردة(Static single Index):

وهي ثلاثة انواع:

1. الفهارس الرئيسية(Primary Index).
2. الفهارس الثانوية(Secondary Index).
3. الفهارس العنقودية(clustering).

وسنأخذها بشيء من التفصيل.

1. الفهارس الرئيسية(Primary Index).

هي عبارة عن فهرس مبني على أساس ملف منظم ومرتب ترتيباً تصاعدياً بالنسبة للمفتاح الرئيسي(Primary Key) ويحتوي الفهرس على حقلين:

a) Indexing field.
b) A pointer to block that includes the record.

وهناك شكلين له:

أ- فهارس غير مضغوطة(Non-densew54t6): كل مؤشر يؤشر على كتلة واحدة.
ب- فهارس مضغوطة: كل مؤشر يؤشر على سجل واحد.

مثال:

لنفرض أن لدينا ملف بيانات يحتوي على 30000 سجل، محفوظ على قرص سعة الكتلة الواحدة فيه 1024 بايت، كل سجل هو من النوع الثابت الغير قابل للتوزيع وبحيث مساحته هي 100 بايت ومع العلم أن سجل الفهرس يحتاج إلى 15 بايت.
احسب كم نحتاج من الوقت في حالة:

1) عدم وجود الفهرس؟
2) عند استخدام الفهرس غير مضغوط؟
3) عند استخدام الفهرس مضغوط؟

الحل:

1) عدم وجود الفهرس؟

$$\Leftarrow \text{عدد السجلات /كتلة البيانات} = \frac{1024}{10} \cong 10 \text{سجلات /كتلة}$$

⇐ عدد الكتلة المطلوبة $= \frac{30000}{10} =$ 3000 كتلة

⇐ باستخدام البحث الثنائي

Estimated Time=Log_2 (3000) =12 block access

2) عند استخدام الفهرس غير مضغوط؟

⇐ عدد السجلات /كتلة الفهرسة= $\frac{1024}{15} \cong$ 68 سجل فهرسة /كتلة

⇐ نحتاج إلى 3000 سجل فهرسة = عدد كتل البيانات.

⇐ عدد الكتلة المطلوبة للفهرس= $\frac{3000}{68} =$ 45 كتلة

⇐ باستخدام البحث الثنائي

Estimated Time= Log_2 (45) +1 =6+1=7 Block Access

3) عند استخدام الفهرس مضغوط؟

⇐ نحتاج إلى 30000 سجل فهرسة = عدد سجلات البيانات.

⇐ كل كتلة فهرسة = 68 سجل فهرسة.

⇐ عدد الكتلة المطلوبة للفهرس= $\frac{30000}{68} =$ 442 كتلة.

⇐ باستخدام البحث الثنائي

Estimated Time= Log_2 (442) =9 Block Access

⇐ فكما نرى استخدام الفهرس أسرع عملية البحث إلى لضعف تقريبا(12-7)، و الفهر س الغير مضغوط أسرع من المضغوط.

2. الفهارس الثانوية(Secondary Index).

هو عبارة عن ملف مرتب، و يحتوي حقلين، الحقل الأول هو عبارة عن أحد حقول البيانات الموجودة و الغير مرتبة، و الحقل الثاني هو عبارة عن مؤشر على كتلة(غير مضغوط) أو سجل (مضغوط).

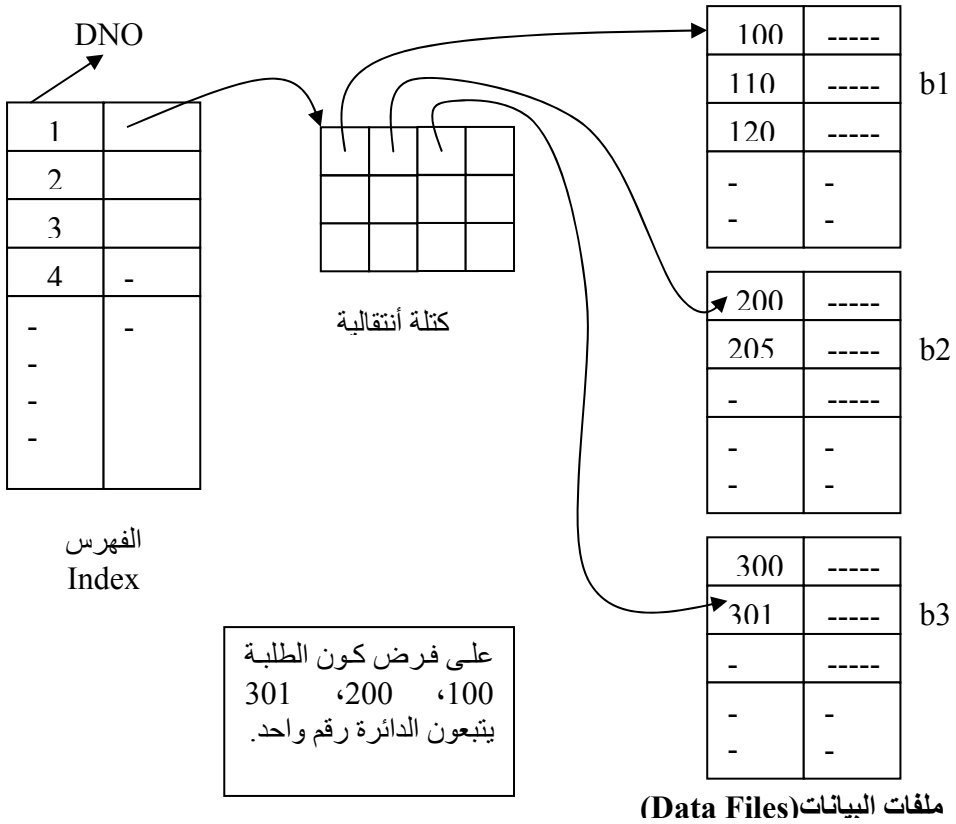
وهذا النوع من الفهارس يأتي على حالتين، تبعاً لنوع حقل الفهرسة:

1. بدون تكرار(Not-repeated):

- مثل استخدام اسم الموظف كحقل فهرسة، فهو حقل لا يتكرر، ولكن ليس هو PK، ويعامل تقريباً نفس معاملة الفهرس الرئيسي.

2. وجود تكرار(repeated):

- مثل الرقم للدائرة بالنسبة للطالب، حيث يكون أكثر من طالب في دائرة واحدة، ونريد أن نبحث بحسب رقم الدائرة.
- ونحل مشكلة التكرار باستخدام فكرة الكتل الانتقالية كما هو موضح في الشكل التالي:



الأفضليات و السيئات للفهرس

- **الأفضليات(Advantages):**

(1) عملية البحث باستخدام البحث أسرع.

- **السيئات(Disadvantages):**

(1) التجديد في ملفات البيانات يحتاج إلى التجديد أيضاً في الفهرس.

(2) ضياع في المساحة.

أسئلة الوحدة:

س[1]) بين ما هو التعريف لكل من المفاهيم التالية :

4) المزج الخارجي.

5) الفهرس.

س[2]) لنفرض أن لدينا ملف بيانات يحتوي على 55000 سجل محفوظة على قرص سعة الكتلة الواحدة فيه هي(1024 byte) , كل سجل هو من النوع الثابت الغير قابل للتوزيع , و بحيث مساحته هي(50 byte) مع العلم أن سجل الفهرس يحتاج إلى (20 byte).

أحسب كم نحتاج من الوقت في حالة:

1) عدم استخدام الفهرس(index)؟

2) باستخدام الفهرس من النوع المضغوط(Dense)؟

س[3]) قارن بين الملفات المرتبة و الغير مرتبة كطريقتين لتنظيم الملفات؟

الوحدة الثالثة: تصميم قواعد البيانات العلائقية (Relational Database Design)

الأهداف العامة للوحدة:

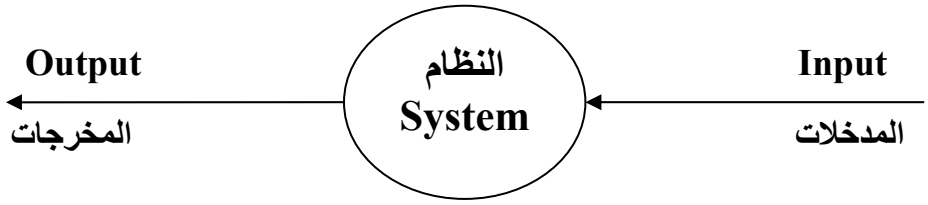
أن يتعرف الطالب/ة على:

1. خطوات تصميم قواعد البيانات.
2. نموذج العلاقات و الكينونات.
3. رسم نموذج.
4. الضوابط البنائية.
5. النموذج العلائقي.
6. ضوابط قواعد البيانات العلائقية.
7. تحويل نموذج العلاقات و الكينونات إلى النموذج العلائقي.

خطوات تصميم قواعد البيانات(Database Design Steps)

يمر التصميم بالخطوات التالية:

(1) جمع المتطلبات و تحليلها (Requirements collection and analysis).



(2) التصميم المنطقي لقاعدة البيانات (Conceptual DB Design)، ويتم بإحدى الطرق التالية:

- النموذج العالي المستوى، وهو نموذج العلاقات و الكينونات (E-R Model). ويتميز بالسهولة في العمل ولكن لا يعطي تصميما مطبعاً.
- باستخدام وسيلة التطبيع(Normalization)، و يتميز بصعوبة العمل، ولكن يعطي تصميماً مطبعاً، أي بدون وجود مشاكل.

(3) التنفيذ الفعلي لقاعدة البيانات باستخدام إحدى أنظمة قواعد البيانات التجارية. و الذي يشمل:

- التحويل من التصميم المنطقي إلى النموذج العلائقي(Mapping to the Relational model).
- تلبية المتطلبات الفعلية: و التي تشمل النماذج (Forms)، و الاستعلامات الدائمة(Queries)، وواجهات التطبيق(Interfaces).

(4) التصميم الفيزيائي لقاعدة البيانات(Physical DB Design)، و الذي يشمل تصميم طرق تنظيم الملفات. و تراكيب الوصول التي منها الفهارس(Indexes).

- لقد غطينا النقطة رقم (4) في الوحدة الثانية وسنبدأ بتغطية النقطة (2) في هذه الوحدة. وعلى فرض وجود أفكار النقطة الأولى عند الطالب.

نموذج العلاقات و الكينونات (E-R Model)

- E-R Model : Entity Relationships Model
- الذي يتكون من :

(1) الكيانات(Entities) والتي لها مجموعة من الخصائص(Properties or Attributes).

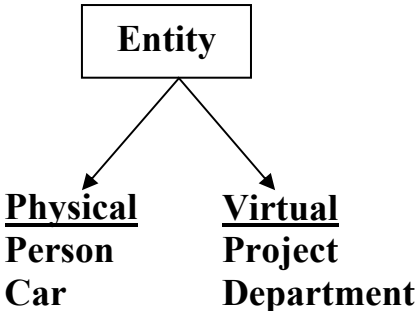
(2) العلاقات المشتركة(Relationships) بين هذه الكيانات.

- وسنقوم بسرد بعض التعريفات اللازمة للتعامل مع نموذج العلاقات الكينونات.

تعريفات:

1. الكيانات(Entities).

- الكيان هو عبارة عن شئ موجود في الواقع، ويتميز بوجود مستقل. و الكيانات نوعان : افتراضية(Virtual)، و حقيقية (physical). كما هو موضح في الرسم المرافق التالي:



- كل كيان يوصف بمجموعة من الخصائص(Attributes).

- مثال كيان الموظف(Employee)، له الخصائص: الأسم(Ename)، رقم الهوية(ENO)، العمر(Age)، الراتب(Salary).

2. نوعية كيان(Entity Type):

- هي عبارة عن مجموعة من الكيانات لها نفس الخصائص.
- مثال: نوعية الكيان الموظف(Employee)، يتبعها الكيانات التالية:

e_1(A,20,10,----------)
e_2(B,----------)
e_3(C,----------)
-
-
e_n(----------)

بحيث أن $e_1,e_2,e_3,-----,e_n$ تتبع نوعية الكيان موظف(Employee).

3. مجموعة كيان(Entity Set): العناصر

- هي عبارة عن مجموعة من الكيانات الإفرادية(single Entities) التي تتبع مجموعة من حالات حصول النوع.
- مثال: في المثال السابق $e_1,e_2,e_3,-----,e_n$ تمثل مجموعة الكيان موظف (Employee).

4. الخصائص المفتاحية(Key Attributes):

- هي الخصائص التي تتميز بوجود قيمة فريدة (Unique) في مجموعة الكيان.
- مثال: في المثال السابق رقم هوية الموظف(ENO) هو خاصية مفتاحية لأنه:

$$e_1(ENO) \neq e_2(ENO) \neq e_3(ENO) \neq - - - \neq e_n(ENO)$$

5. أنواع الخصائص(Types of Attributes)

(1) خصائص إفرادية القيمة(Single Value):

مثل: تاريخ الميلاد، العمر

(2) خصائص متعددة القيمة(Multi valued):

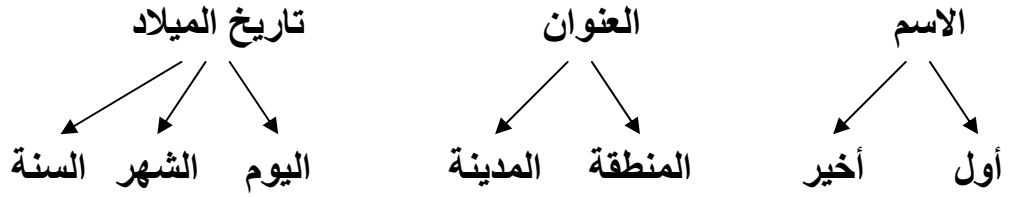
مثل: {الأماكن}، {اللون}

(3) خصائص بسيطة أو نووية(simple or atomic):

مثل: العمر

(4) خصائص مركبة(Composite):

مثل:

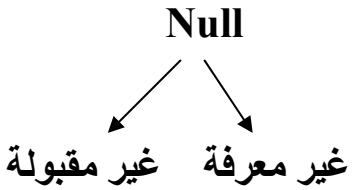


(5) خصائص مشتقة(derived):

مثل:العمر(التاريخ الحالي-التاريخ للولادة)

(6) خصائص قد لا يكون لها قيمة(May Be NULL):

مثل: رقم الهاتف



6. مجال الخاصية(Domain of Attribute)

- هي عبارة عن مجموعة القيم التي من الممكن أن تستخدم كقيمة للخاصية.
- مثال: الراتب للموظف (salary) يتراوح بين (1000شيكل و 5000 شيكل).

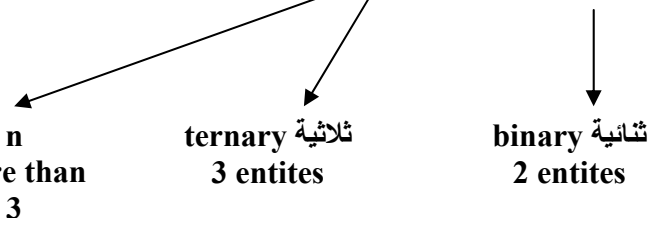
7. العلاقات المشتركة(Relationships)

- العلاقة المشتركة R بين n كيانات تُعَرِف مجموعة من الارتباطات بين هذه الكيانات.
- R= مجموعة من حالات الحصول (r_i).

R= set of Instances (r_i)

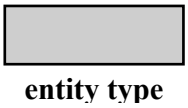
8. الدرجة للعلاقة المشتركة(Relationship Degree):

الدرجة للعلاقة المشتركة= عدد الكيانات المشتركة في العلاقة

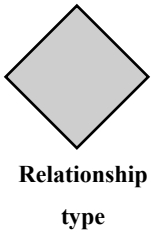


رسم نموذج E-R Model:

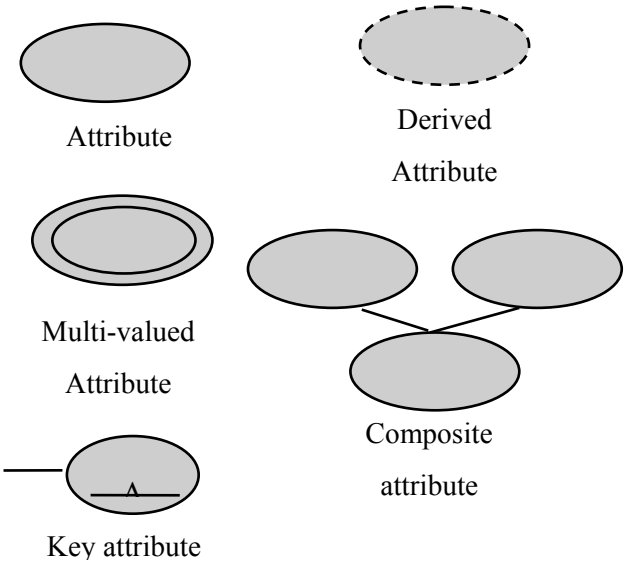
نرمز للكيان بالشكل التالي:



وللعلاقة المشتركة بالشكل التالي:

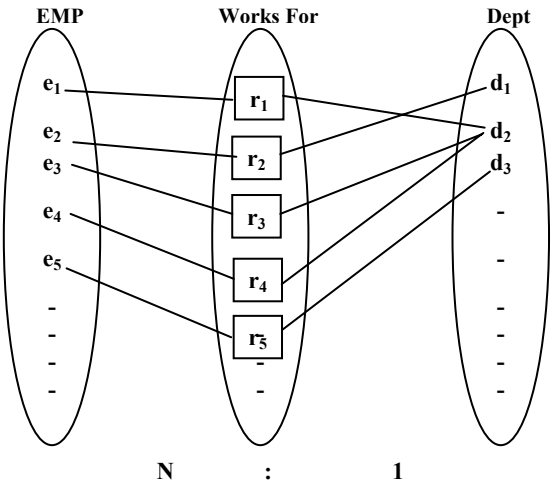


بعض الرموز الأخرى المستخدمة في :E-R Model

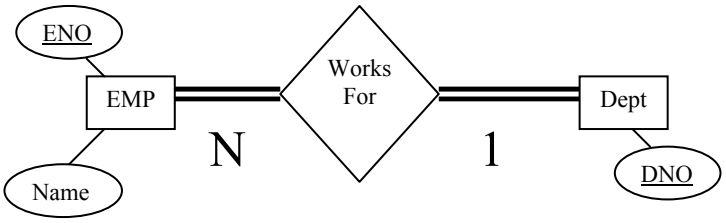


بالنسبة للمثال الموجود في الملحق(قاعدة بيانات لشركة): عندنا ثلاث حالات:

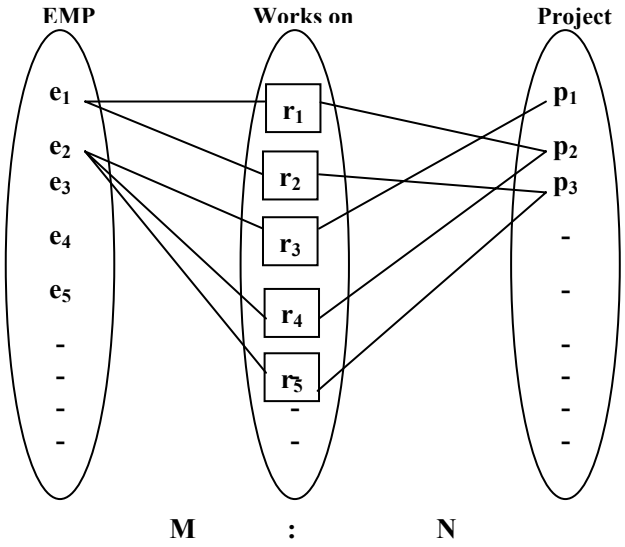
(1) الحالة الأولى:



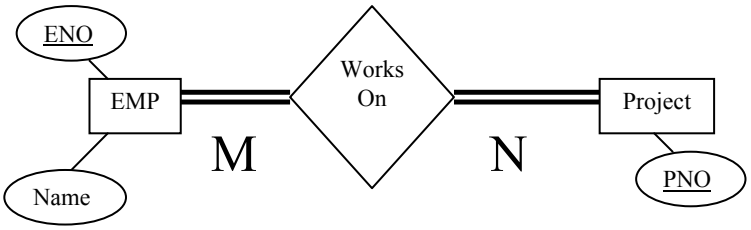
ونعبر عنها بالشكل التالي



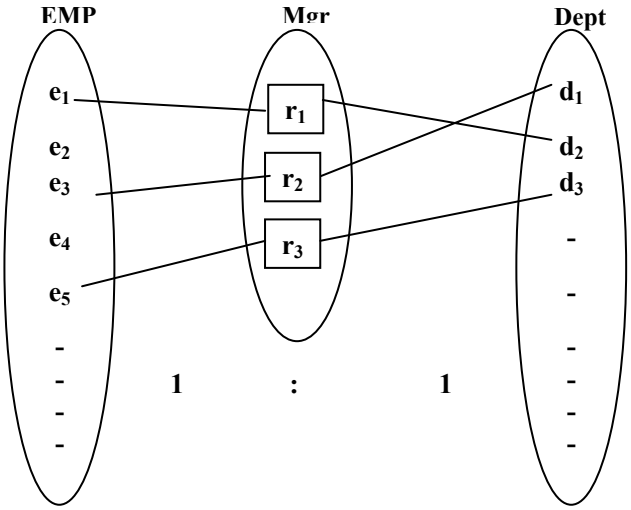
(2) الحالة الثانية:



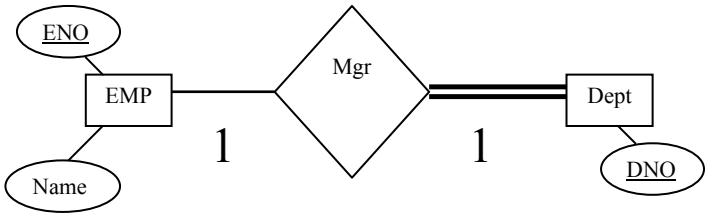
ونعبر عنها بالشكل التالي



(3) الحالة الثالثة:



- ونعبر عنها بالشكل التالي



الضوابط البنائية (Structural Constraints)

- وهي نوعان:

1. Cardinality Ratio(نسبة الجوهرية للعلاقة المشتركة):

- وتعرف بأنها عدد حالات الحصول للعلاقة المشتركة التي يستطيع الكيان الارتباط بها.
- ويمكن أن تكون أحدى الحالات التالية:

1:1	واحد-لواحد	رأس برأس
1:N	واحد-لمتعدد	رأس بأطراف
M:N	متعدد-متعدد	أطراف بأطراف
N:1	متعدد-واحد	أطراف برأس

2. Participation(الارتباط):

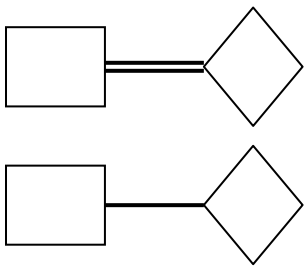
- وتعرف بأنها تبيان هل أن وجود الكيان يعتمد على وجود كيان أخر بواسطة العلاقة المشتركة.
- وقد تكون كاملة أو جزئيه:

(1)كاملة(FULL):

وتعني أنه إذا كان هنالك كيانان S, F : كل حالة حصول للكيان (F) يجب أن ترتبط بحالة حصول للكيان (S)

(2)جزئية(Partial):

و تعني أنه ليس شرطا أن يكون لكل حالة حصول للكيان (F) أن ترتبط بحالة حصول للكيان (S)



كاملة(FULL)

جزئية(Partial)

نوعية الكيانات الضعيفة(WE: Weak Entity type)

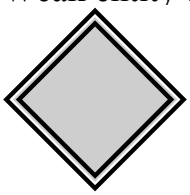
- هي الكيانات التي تكون مرتبطة بنوعية كيان معينة أخرى ويكون هذا الارتباط من خلال بعض الخصائص.

- نوعية الكيان الأخرى نسميها الكيان المالك أو المفسر(OE: Owner Entity).
- العلاقة المشتركة بين الكيان الضعيف و الكيان المفسر تسمى العلاقة المفسرة(IR: Identifying Relationship).
- كل حالة حصول للكيان الضعيف تعتمد في وجودها على حالة حصول للكيان المفسر(OE).
- المفتاح للكيان الضعيف لا يسمى مفتاح رئيسي ولكن يسمى المفتاح الجزئي(Partial Key).

- **نرمز للكيان الضعيف بالشكل التالي:**

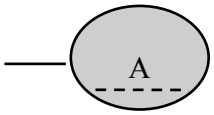
Weak entity type

- **وللعلاقة المشتركة المفسرة بالشكل التالي:**



IR

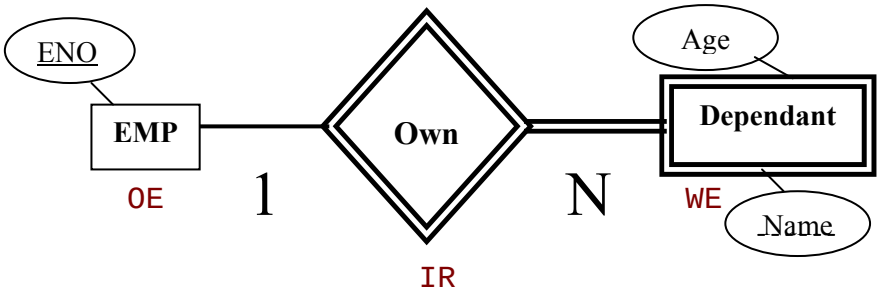
- **و للمفتاح الجزئي بالرمز:**



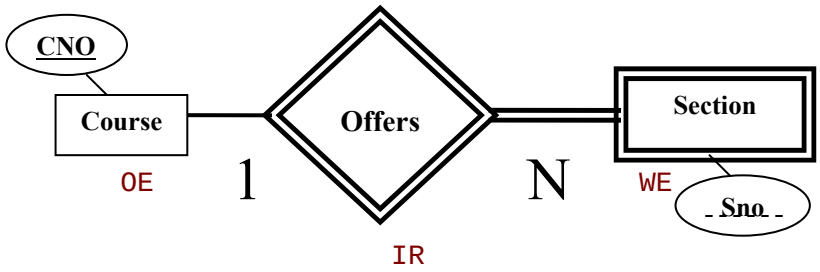
Partial Key attribute

مثال:

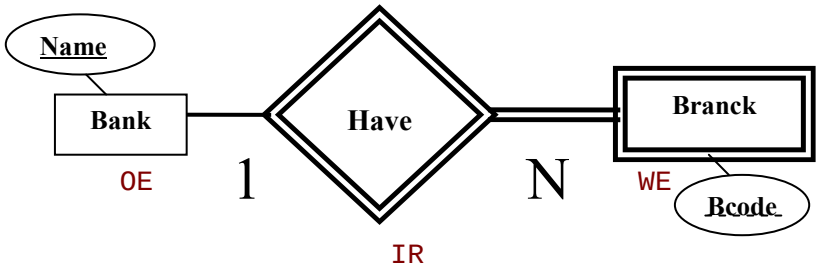
(1)كل موظف(EMP) يمكن أن يكون له معالين(Dependant)، ولكن كل معال يجب أن يكون تابعاً(Weak) لموظف معين:



(2)كل مساق(Course) يمكن أن يكون له شعب(Section)، ولكن كل شعبة يجب أن تكون تابعةً(Weak) لمساق معين:



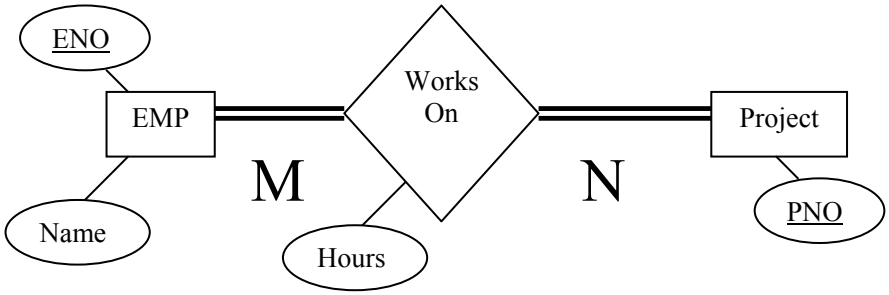
(3)كل بنك(Bank) يمكن أن يكون له فروع(Branch)، ولكن كل فرع لبنك يجب أن يكون تابعاً(Weak) لبنك معين:



- كما نلاحظ فأن العلاقة المفسرة هي من نوع(1:N) و لها ارتباط كامل من جهة الكيان الضعيف.

ملاحظة على نموذج العلاقات و الكينونات :

بعض العلاقات المشتركة قد يكون لها خصائص، مثل ان الموظف يعمل على مشروع معين بعدد ساعات معينة(Hours):



مثال كامل على نموذج العلاقات الكينونات(قاعدة البيانات لجامعة)

- بالأخذ بعين الاعتبار المجموعة التالية من المتطلبات لقاعدة البيانات لجامعة، التي تستخدم لمتابعة سجلات الطلبة:

(1)بالنسبة للطالب(student)، الجامعة تهتم بالبيانات التالية:

Student name, Number, Social Security Number (SSN), current address and phone, permanent address and phone, birth date, sex, Class(freshman,

Sophomore,---,graduate), major department, minor department(if any), and degree program (B.A, B.S, ---, PhD).

مع العلم أن بعض برامج التطبيق تحتاج للوصول إلى المدينة أو الدولة أو الرمز البريدي للطالب، و أيضاً الاسم الأخير للطالب. أيضاً نعرف بان SSN, Student Number لها قيم فريدة لكل طالب.

(2) بالنسبة للدائرة(Department)، توصف بالتالية:

Name, Department code, Office number, office phone, and collage.

أيضاً نعرف بان Code, name لها قيم فريدة لكل دائرة.

(3) بالنسبة للمساق(Course)، توصف بالتالية:

Course Number, Number of Semester Hours, level, offering department.

أيضاً نعرف بان Course number لها قيم فريدة لكل مساق.

(4) بالنسبة للشعبة(Section)، توصف بالتالية:

Instructor, Semester, Year, Course, and Section Number.

أيضاً نعرف رقم الشعبة يفرق بين الشعب المختلفة للمساق المعني الذي يدرس في فصل و سنة معنية.

(5) تقرير العلامات للطالب، يحتوي على طالب، شعبة، وعلامة كتابية(جيد،جيد جداً، ممتاز)، وعلامة رقمية(99،88،----).

المطلوب:

صمم نموذج العلاقات و الكينونات لهذا التطبيق، وأرسمه، عين الخصائص المفتاحية، لكل نوعيات كيانات، و الضوابط البنائية على كل علاقة مشتركة، لاحظ وجود أي متطلبات غير مذكورة و اعمل ما هو مطلوب ليكون هذا الوصف كاملاً:

خطوات الحل:

أ- تحليل البيانات.

ب- استخراج الكيانات و الخصائص(Entity Type, Attrbutes):

1. Student:

   Sname, SNO, SSN, …….

2. Dept:

Dname, Dcode, office_NO, office_phone, collage.

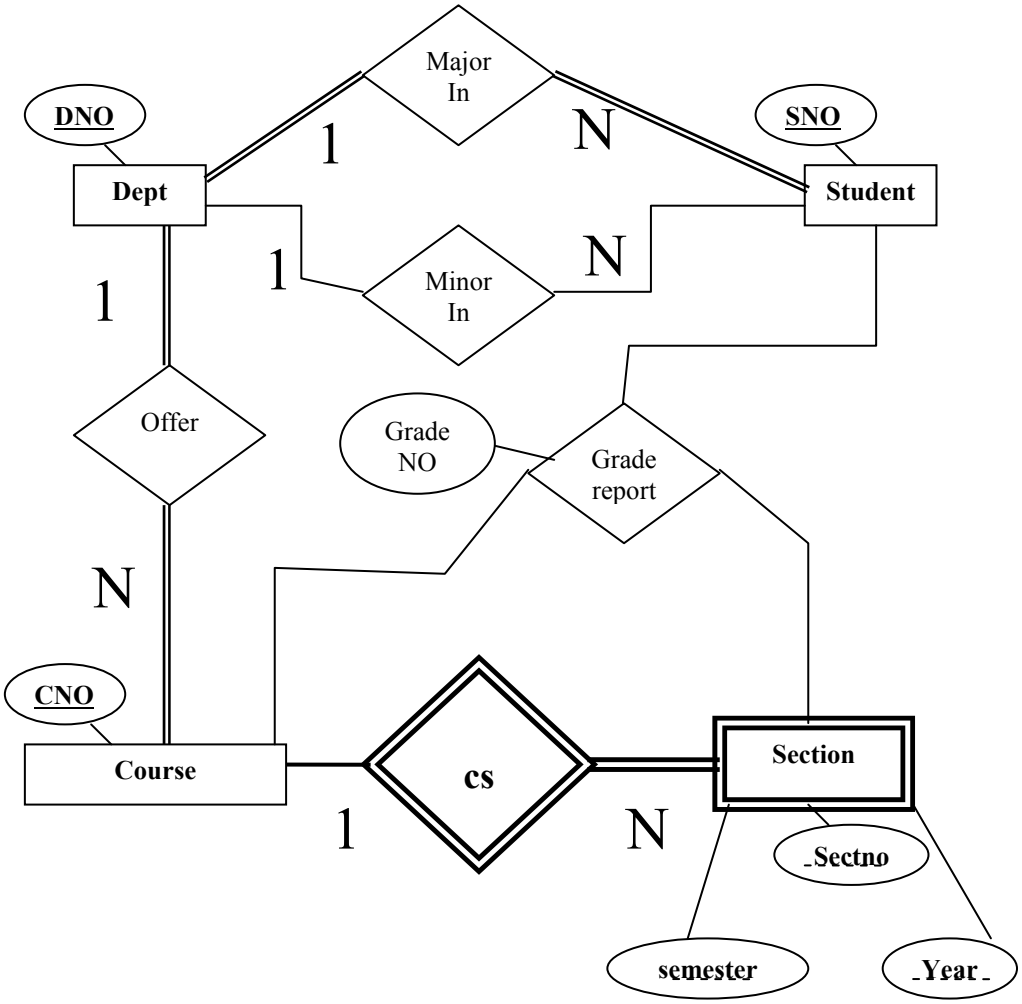
3. Course: OE

   CNO, Hours, Level.

4. Section: WE

   Instructor, semester, year, SECTno.

ج- نرسم نموذج (E-R Model)، الرسمة التالية تمثل جزأ من النموذج، قم برسمه كاملاً على ورقة كتمرين:



النموذج العلائقي(Relational Model)

- النظام العلائقي هو الأساس لقواعد البيانات العلائقية(مجموعة من الجداول وعلاقات بينها).
- مخترع النظام العلائقي هو كود(Codd) في عام 1970.

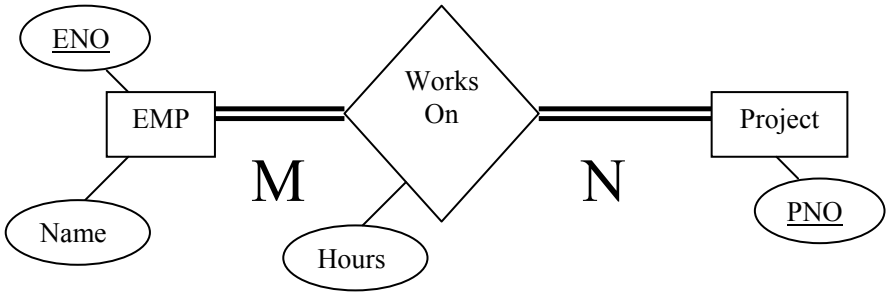
تعريفات:

1. العلاقة (Relation):

- هي عبارة عن حقيقة(Fact) تمثل كيان أو علاقة مشتركة(Relationship).
- أمثلة:

(1) الطالب: Student(No, Name, Address,…).

(2) العلاقة المشتركة يعمل على(Works on):



تحول إلى علاقة : Works_on(Eno,Pno,Hours).

- العلاقة (Relation): هي تمثل الجدول (Table)، بتحولات المفاهيم التالية:

Attribute(خاصية)	→	Column Headers(عمود رأسي).
Tuple(عنصر)	→	Row(صف).
Domain(المجال)	→	Data Type(نوعية البيانات).

- العلاقة مكونة من التالية:

(1) رأس (Heading):

يتكون من مجموعة من الحقول (A_1 ..... A_n)، بحيث ان كل حقل من الحقول يتبع مجال معين (D_i).

(2) جسم(Body):

يتكون من مجموعة متغيرة مع الزمن من العناصر(Tuples) أو الصفوف (Rows)، حيث أن كل صف يتكون من مجموعة من الخصائص.

- خصائص العلاقة:

كما قلنا، العلاقة هي مجموعة (Set)، و تأخذ جميع الخصائص للمجموعة الرياضية التالية:

(1) لا يشترط في العلاقة ترتيب العناصر(Tuples).

(2) الحقول للعنصر لا يشترط فيها الترتيب.

(3) لا يجوز تكرار العناصر.

(4) كل عنصر يحتوي على قيمة واحدة لكل حقل، لا يجوز أن يكون هنالك حقول متعددة القيمة.

2. المجال(Domain):

- المجال هو عبارة عن مجموعة من القيم الافرادية النووية، بحيث أن كل هذه القيم من نفس النوع، وكل مجال له أسم مجال(Domain Name) و نوعية بيانات(Data Type) و شكل (Format):

Each Domain → Name, Data Type, Format.

- مثال:

رقم الطالب: رقم صحيح، 1 حتى 999999، أول رقمين يمثلان السنة والبقية الطالب في هذه السنة.

- المجال المركب(Composite domain):

- مجموعة مركبة من عدة مجالات بسيطة(Collection of Simple Domains)، مثال:

تاريخ الميلاد: Bdate(Day(1-31), Month(1-12), Year) .

3. درجة العلاقة(Relation Degree):

- تمثل عدد الخصائص الحقول في العلاقة و تكون عادة ثابتة مع الزمن:

No of attributes in the relation, fixed with time.

4. جوهرية العلاقة (Relation Cardinality):

- تمثل عدد العناصر في العلاقة و تكون عادة متغيرة مع الزمن:

No of Tuples (Rows) in the relation, Varies with time.

5. قاعدة البيانات العلائقية(Relational DB)

قاعدة بيانات تتكون من مجموعة متغيرة مع الزمن من العلاقات(Relations) ذات درجة علاقة معينة، أنها تعتمد على النموذج العلائقي و الذي يعتبر نموذج بيانات منطقي:

Relational DB= S { R_1, R_2, R_3,----------, R_n}

بحيث أن لكل من (R_n1R ,…..) خطة علائقية (حقول) ، مع مجموعة من ضوابط صحة المعلومات (Integrity Constraints).

ضوابط قواعد البيانات العلائقية (Relational Db Constraints)

1. ضوابط المجال(Domain Constraints)

تتمثل بان كل قيمة لكل حقل A يجب ان تكون قيمته تتبع مجال معين D، و كل مجال يجب أن يتبع نوعية بيانات معينة(Data Type).

2. ضوابط المفاتيح(Key Constraints):

تتمثل هذه الضوابط في الشيئيين التاليين:

أ- المفاتيح الرئيسية (PK: Primary Key).

ب- المفاتيح الغريبة (FK: Foreign Key).

أ- المفاتيح الرئيسية (PK: Primary Key).

- في العلاقة R :

Relation R { A_1, A_2, A_3,----------, A_n}

- تعتبر المجموعةK التي هي مجموعة فرعية من R:

R { A_i, A_j, ----------, A_k}

تعتبر مفتاح مرشح (Candidate key) للعلاقة R إذا حققت الشروط التالية:

(1) قاعدة الإفرادية(Uniqueness Rule):

أي الحقول لها قيم فريدة، و بالشكل الرياضي:

$$T_i[S_k] \neq T_j[S_k] \quad For \quad All \quad i \neq j$$

أذا كان الحقل أو مجموعة الحقول (S_k) تحقق قاعدة الأفرادية ، يكون المفتاح فائق (Super key).

(2) قاعدة التقليل(Minimality Rule):

تعني حذف أي حقل A من المفتاح الفائق، يترك مجموعة من الحقول التي ليست مفتاح للعلاقة.

- إذا كان الحقل أو مجموعة الحقول تحقق قاعدة الإفرادية و التقليل يسمى مفتاح مرشح(Candidate Key).
- في النهاية سيكون عندنا مجموعة من المفاتيح المرشحة، يجب ان نختار أحداها كمفتاح رئيسي للعلاقة R.

ب- المفتاح الغريب(Foreign Key):

- حقل أو مجموعة من الحقول (Fk) في العلاقة R_1، هي مفتاح غريب للعلاقة R_2، كالتالي:

$$D(FK) \qquad\qquad D(PK)$$
$$R_1 \qquad\qquad R_2$$

(1) الحقول في FK لها نفس المجال للمفتاح الرئيسي في العلاقة الأخرى.

(2) القيم للمفتاح الغريب FK في أحد العناصر T_1 في العلاقة الأولى، يجب أن تكون أما:

أ. قيم للمفتاح الرئيسي (PK) في العلاقة R_2 في أحدى العناصر T_2.

ب. بدون قيمة (NULL Value).

و بالشكل الرياضي:

$$T_1[FK] = T_2[PK]$$
$$\Rightarrow T_1[R_1] \xrightarrow[\text{Re } fers]{} T_2[R_2]$$
$$or\ T_1[FK] = NULL$$

3. القواعد الضبطية (Constraints Rules):

أ- صحة الكيان (Entity Integrity):

لا يجوز أن تكون أحد الحقول في المفتاح الرئيسي بدون قيمة (NULL).

ب- صحة التوزيع (Referential Integrity):

في العلاقة R_2 التي تحتوي على مفتاح غريب(FK)، الذي يمثل مفتاح رئيسي (PK)، في العلاقة R_1 ، يجب أن تكون كل قيمة للمفتاح الغريب إما:

1. تساوي قيمة المفتاح الرئيسي في إحدى العناصر للعلاقة R_1.
2. أو بدون قيمة(NULL).

تحويل نموذج العلاقات و الكينونات إلى النموذج العلائقي (ER To Relational)

يتم تحويل نموذج العلاقات و الكينونات الى النموذج العلائقي عبر الخطوات التالية:

- **الكيانات العادية(Regular Entities):**
  - كل كيان عادي S يحول إلى جدول R. و نجعل كل الخصائص البسيطة و الأقسام البسيطة للخصائص المركبة كحقول في الجدول R.
  - المفتاح المرشح للجدول الناتج R هو المفتاح الرئيسي للكيان S.

- **الكيانات الضعيفة(Week Entities):**
  - كل كيان ضعيف S مملوك للكيان المالك T يحول إلى جدول R. و نجعل كل الخصائص البسيطة و الأقسام البسيطة للخصائص المركبة كحقول في الجدول

R. كما ونجعل كل الخصائص البسيطة و الأقسام البسيطة للخصائص المركبة للعلاقة المشتركة المعرفة(IR) كحقول في الجدول R.

- المفتاح المرشح للجدول الناتج R هو عبارة عن مركب من المفتاح الجزئي للكيان الضعيف S مع المفتاح الرئيسي للكيان المالك T.
- نجعل الجزء الممثل للكيان المالك في المفتاح المرشح مفتاحاً غريباً يعود على المفتاح الرئيسي للجدول الذي يمثل الكيان المالك.

- **العلاقات المشتركة الثنائية من نوع متعدد الى متعدد(Many-Many Relationships):**
  - كل علاقة مشتركة من نوع متعدد إلى متعدد S بين الكيانين (T1,T2) تحول إلى جدول R. كما ونجعل كل الخصائص البسيطة و الأقسام البسيطة للخصائص المركبة للعلاقة المشتركة S كحقول في الجدول R.
  - المفتاح المرشح للجدول الناتج R هو عبارة عن مركب من المفتاحين الرئيسيين للكيانين المشتركين في العلاقة.
  - نجعل الجزئين الممثلين للكيانين المشتركين في العلاقة في المفتاح المرشح مفتاحين غريبين يعودان على المفتاح الرئيسي للجدول الذي يمثل كل واحد منهما.

- **العلاقات المشتركة الثنائية من نوع واحد الى متعدد(One-Many Relationships):**
  - كل علاقة مشتركة من نوع واحد إلى متعدد S بين الكيانين (T1,T2)، بحيث أن T1 من جهة الواحد و T2 من جهة متعدد، وبحيث أن (R1,R2) الجدولين الذين يمثلان الكيانين(T1,T2).
  - نجعل المفتاح الرئيسي للجدول R1 حقلاً في الجدول R2. ونجعل هذا الحقل (أو مجموعة الحقول) مفتاحاً غريباً يعود على المفتاح الرئيسي للعلاقة R1. كما ونجعل كل الخصائص البسيطة و الأقسام البسيطة للخصائص المركبة للعلاقة المشتركة S كحقول في الجدول R2.

- **العلاقات المشتركة الثنائية من نوع واحد الى واحد(One-One Relationships):**
  - كل علاقة مشتركة من نوع واحد إلى واحد S بين الكيانين (T1,T2)، وبحيث أن (R1,R2) الجدولين الذين يمثلان الكيانين(T1,T2).
  - ننظر إن كان الارتباط كامل من أحدى الجهات فقط مثلاً R1. نجعل المفتاح الرئيسي للجدول R2 حقلاً في الجدول R1. ونجعل هذا الحقل (أو مجموعة الحقول) مفتاحاً غريباً يعود على المفتاح الرئيسي للعلاقة R2. كما ونجعل كل

الخصائص البسيطة و الأقسام البسيطة للخصائص المركبة للعلاقة المشتركة S كحقول في الجدول R1.

- أما أن كان الارتباط جزئي أو كامل من الجهتين. نكون مخيرين في اختيار المفتاح الغريب من R1 أو R2. ونقوم بنفس الشيء في الخطوة السابقة.

- **العلاقات المشتركة الثلاثية أو أكثر:**

- كل علاقة مشتركة ثلاثية أو أكثر S بين الكيانات (T1…Tn) تحول إلى جدول R. كما ونجعل كل الخصائص البسيطة و الأقسام البسيطة للخصائص المركبة للعلاقة المشتركة S كحقول في الجدول R.
- المفتاح المرشح للجدول الناتج R هو عبارة عن مركب من المفاتيح الرئيسية للكيانات المشتركة في العلاقة.
- نجعل الأجزاء الممثلة للكيانات المشتركة في العلاقة في المفتاح المرشح مفاتيح غريبة تعود على المفتاح الرئيسي للجدول الذي يمثل كل واحد منها.

- **الخصائص متعددة القيمة(Multi-valued Attributes):**

- الخاصية متعددة القيمة s في الكيان S. تحول إلى جدول R يتكون من حقلين هما الخاصية بالإضافة إلى المفتاح الرئيسي للجدول الذي يمثل الكيان S كمفتاح غريب يعود عليه.
- المفتاح المرشح للجدول الناتج R هو عبارة عن مركب من الحقلين.

أسئلة الوحدة:

س[1]) بين ما هو التعريف لكل من المفاهيم التالية :

1. الكيان, الخصائص, العناصر.
2. المجال للخاصية(Domain of attribute).
3. العلاقة (Relation).
4. خاصية متعددة القيمة(Multi valued Attribute).
5. العلاقة المشتركة(Relationship).
6. المفتاح المرشح(Candidate key).
7. صحة التوزيع(Referential Integrity).

س[2])قارن بين الكيان الضعيف(Week Entity) و الكيان العادي(Regular Entity)؟

س[3]) بالأخذ بعين الاعتبار المجموعة التالية من المتطلبات ل قاعدة بيانات لعيادة طبيب(Clinic), التي تستخدم لتتبع وضع المرضى (sick) و الأدوية(Drugs):

- ✓ المريض يُعرَفْ برقم المريض(sick number)، اسم المريض(Sickname)، عمره، جنسه، رقم الهاتف، العنوان. بحيث أن رقم المريض فريد القيمة.
- ✓ الدواء يعرف بالتالية: اسم الدواء(Drug Number) ، رقم الدواء(Drug number), تاريخ الشراء، تاريخ الانتهاء. بحيث أن أسم الدواء و رقمه فريد القيمة.
- ✓ الطبيب(Doctor) و يعرف بالتالية:أسم الطبيب(Doctor name)، رقم الطبيب(Doctor number)، الراتب، تاريخ بداية العمل. بحيث أن رقم الطبيب فريد القيمة.
- ✓ مع العلم أن المريض يعالج عند طبيب واحد للمرض(Disease) الواحد، ويكون له تاريخ أخر مراجعة(Last Visit) . وأيضاً مع العلم أن الطبيب قد يصرف الدواء للمريض، ويأخذ المريض الدواء بوصفة معينة(usage).

المـطـلـــــوب:

بالاعتماد على المتطلبات المذكورة آنفا:

1. صمم نموذج العلاقات و الكينونات(ER-Model) لهدا التطبيق و ارسميه. عيني الخصائص المفتاحية لكل نوعيات كيان, و الضوابط البنائية على كل علاقة مشتركة ، و أضيف ما هو مطلوب لجعل الوصف كاملاً مع ذكر ذلك.
2. حول نموذج أل(ER-Model) إلى النموذج العلائقي(Relational Model)، مع تحديد ضوابط المفاتيح(FK and PK)

س[4]) بالأخذ بعين الاعتبار المجموعة التالية من المتطلبات ل قاعدة بيانات لمكتبة(Library), التي تستخدم لتتبع وضع الكتب(Book) و أعارتها(Loan details):

- ✓ الكتاب يُعرَفْ برقم الكتاب(Book number)، عنوان الكتاب(Title) , و وصف الكتاب(Description)، بحيث أن رقم الكتاب فريد القيمة وفي المكتبة قد يكون عدة نسخ من الكتاب.
- ✓ كل كتاب له ناشر(Publisher) يعرف بالتالية: اسم الناشر(Name) وعنوان(Address)، ورقم الهاتف(Phone Number). بحيث أن أسم الناشر فريد القيمة.

- ✓ كل كتاب له عدة مؤلفين(Authors) على الأكثر، و يعرفون بالتالية:رقم المؤلف(Author ID)، اسم المؤلف(Name) والعنوان البريد الإلكتروني(Email). بحيث أن رقم المؤلف فريد القيمة.
- ✓ المستعير للكتاب(Borrower) يعرف بالتالية: أسم المستعير(Name), عنوان المستعير(Address)، و رقم الهاتف (Phone Number).). بحيث أن أسم الناشر فريد القيمة. مع العلم أن المستعير يستطيع استعارة أكثر من كتاب من المكتبة, لفترة زمنية معينة والتي تبدأ بتاريخ ألاستعارة(Borrowing Date) و تنتهي بتاريخ الإعادة(Date Due).

المـطـلـــــوب:

بالاعتماد على المتطلبات المذكورة آنفا:

1. صمم نموذج العلاقات و الكينونات(ER-Model) لهدا التطبيق و ارسميه. عيني الخصائص المفتاحية لكل نوعيات كيان, و الضوابط البنائية على كل علاقة مشتركة ، و أضيف ما هو مطلوب لجعل الوصف كاملاً مع ذكر ذلك.
2. حول نموذج أل(ER-Model) إلى النموذج العلائقي(Relational Model)، مع تحديد ضوابط المفاتيح(FK and PK)

الوحدة الرابعة: التطبيع(Normalization)

الأهداف العامة للوحدة:

أن يتعرف الطالب/ة على:

1. الإعتمادات الوظيفية
2. نماذج التطبيع.
3. مفهوم التطبيع
4. نماذج التطبيع الموجودة:
   (1) نموذج التطبيع الأول.
   (2) نموذج التطبيع الثاني.
   (3) نموذج التطبيع الثالث.
   (4) نموذج تطبيع بويس كود.
   (5) نموذج التطبيع الثاني(NF2) حسب بويس كود.
   (6) نموذج التطبيع الثالث(NF3) حسب بويس كود.
   (7) نموذج بويس كود المعدل.

الإعتمادات الوظيفية (FD: Functional Dependencies)

- هي عبارة عن ضابط بين مجموعتين من الحقول:
- **مثال: على فرض وجود العلاقة R:**

R $\{A_1, A_2, \ldots\ldots\ldots, A_n\}$

R

A1	A2	.....	An

- **الاعتماد الوظيفي : $X \rightarrow Y$ ، يعني**
  - **X تؤدي إلى Y.**
  - **X تحدد قيمةY أو أن Yتعتمد على X.**
  - **X, Y مجموعة جزئية من R ($R \supseteq X, Y$)**
- **X تؤدي إلىY يعني أنها عبارة عن ضابط على العناصر المحتملة التي تمثل حالة حصول للعلاقة.**
- **وبالشكل الرياضي:**

لأي من العناصر T1, T2 في Rبحيث أن:
T1 [x] =T2 [x]
يجب أن يكون هنالك T1[y]=T2[y]
بحيث أن $X \rightarrow Y$

سؤال :

متى يكون الاعتماد كاملا؟

الحل:

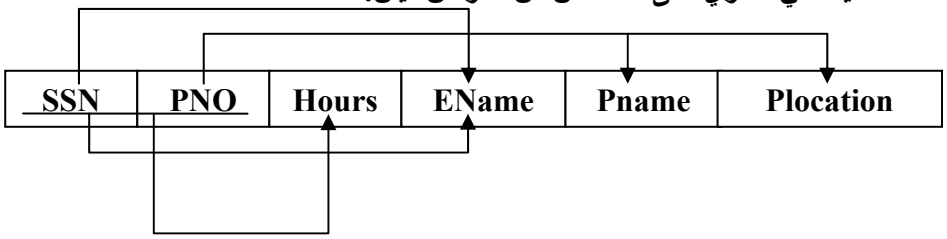
في حالة وجود المفتاح الرئيسي كمحدد.

استنتاجات:

- إذا كان $X \to Y$ فلا يعني ذلك أن $Y \to X$.
- إذا كانت X مفتاح مرشح (Candidate Key)، وكانت $X \to Y$ ، فإن Y يمكن أن تكون أي مجموعة جزئية في R .

مثال:

لدينا العلاقة التالية التي تحتوي على خصائص من أكثر من كيان:



$FD1 : SSN \to Ename$
$FD2 : PNO \to Pname, Plocation$
$FD3 : SSN , PNO \to Hours$
$FD4 : SSN , PNO \to Ename$

ملاحظة : FD3 إعتماد كلي .
: FD4 إعتماد جزئي .

سؤال:

كيف يمكن استخراج الاعتمادات الوظيفية في العلاقة؟

إما باستخدام المتطلبات الموجودة أو كان لدينا مجموعة من العناصر أو البيانات تستخرج منها الإعتمادات الوظيفية، و يجب أن يتوفر التالية في $X \to Y$:

(1) X يجب أن تؤدي إلى Y في كل العناصر .

(2) إذا كانت X عبارة عن مفتاح رئيسي أو مفتاح مرشح للعلاقة R إذن X تؤدي إلى كل العناصر:

$X \to$ all other attributes ;

ولذلك نسمي X في هذه الحالة المحدد(Determinate).

نماذج التطبيع

تعريف التطبيع:

إن العلاقة تسمى علاقة مطبعة (normalized) إذا حققت الشروط لنموذج تطبيع.

نماذج التطبيع الموجودة:

1. **نموذج التطبيع الأول (1NF:First Normal Form).**
2. **نموذج التطبيع الثاني(2NF).**
3. **نموذج التطبيع الثالث(3NF).**
4. **بويس كود(BCNF).**
5. **نموذج التطبيع الرابع (4NF).**
6. **نموذج التطبيع الخامس(5NF).**

نقول علامة مطبعة على النموذج الأول إذا حققت شروط النموذج الأول. و العلاقة مطبعة على النموذج الثاني إذا حققت شروط النموذج الأول و الثاني.وهكذا...

(1)نموذج التطبيع الأول (1NF):

العلاقة تكون محققة لنموذج التطبيع الأول إذا كان كل عنصر في العلاقة(الجدول) يحتوي على قيمة واحدة لكل حقل.

- No. multi valued attributes.
- No. composite attributes.

مثال:

Dept

DNO	Dname	Major	Dlocations

كما نعرف فأن مواقع الدائرة (Dlocations) هي متعدد القيمة و لذلك هذه العلاقة ليست محققة لنموذج التطبيع الأول، وحتى نحول هذه العلاقة إلى علاقة تنطبق عليه (1NF)، نقوم بعملية التقسيم (Decomposition)، لهذه العلاقة إلى علاقتين كالتالي:

Dept

DNO	Dname	Major	Dlocations

Dept_locations

DNO	Dlocations

مثال :

Sp

S#	P#	City	Status	Qty
S1	p1	London	20	100
S1	p2	London	20	250
S2	p1	Rome	10	50
S2	p3	Rome	10	60
S3	p1	Rome	10	70
S3	p4	Rome	10	130
S4	p1	London	50	100

من خلال دراسة البيانات التالية يمكن الحصول على

FD1: city→status;
FD2: s#, P# → QTY;
FD3: S# → city, status;

هذا المثال ينطبق على نموذج التطبيع الأول.

ملاحظة :مشاكل التحديث: update anomalies

هي تلك المشاكل التي تظهر عند إضافة عناصر جديدة إلى العلاقة (relation) وعمليات التحديث و التي تشمل الإضافة و الحذف و التجديد و لذلك يجب الأخذ بعين الاعتبار تلك المشاكل في تحديد الإعتمادات التطبيقية المناسبة لتلك العلاقة.

(2)نموذج التطبيع الثاني(2NF):

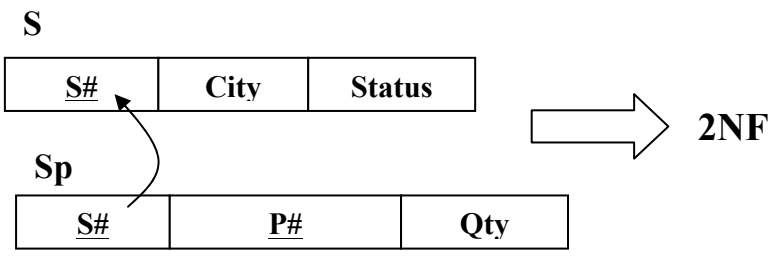
العلاقة تكون على نموذج التطبيع الثاني إذا كان:

1. 2NF يجب أن ينطبق على 1NF.
2. كل حقل غير مفتاحي (non key) غير قابل للتقسيم على المفتاح المرشح لتلك العلاقة .

أي الحقل يعتمد اعتماداً كلياً على المفتاح المرشح.

مثال

- المثال السابق لا ينطبق على هذا النموذج لأن QTY تعتمد فقط على S#,P#
  لكن city , status تعتمد فقط على s# و نحل المشكلة ببناء علاقة جديدة .



الإعتمادات الجديدة:

QTY →S#,P#; (اعتماد كلي).
City, status →S#; (أعتماد جزئي)
City → status;

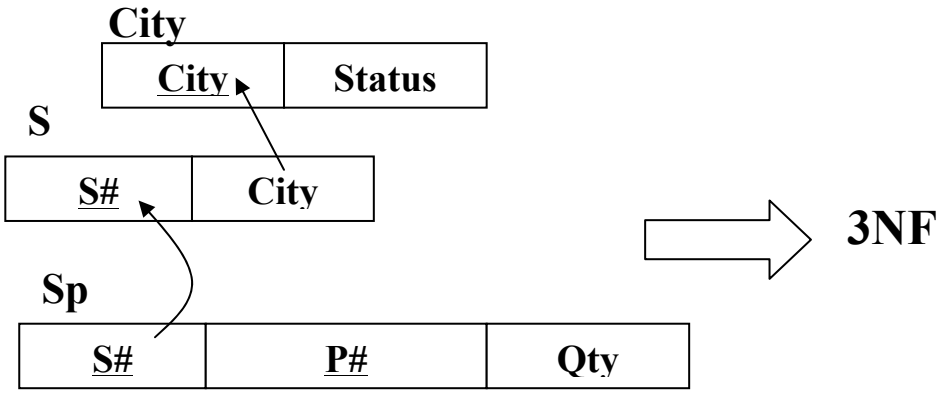
ملاحظة:

التقسيم يجب أن يكون مسترجع (يمكن التراجع عنه) reversible.

(3)نموذج التطبيع الثالث(3NF):

العلاقة R تكون على نموذج التطبيع الثالث إذا:
1. كانت على نموذج التطبيع الثاني.
2. لكل حقل غير مفتاحي (ليس مفتاح رئيسي أو جزء من المفتاح الرئيسي)(Nun prime or Non Key). يجب ألا تكون تعتمد إنتقالياً على المفتاح الرئيسي.

في المثال السابق (يوجد اعتماد انتقالي) تحل هذه المشكلة كالتالي:

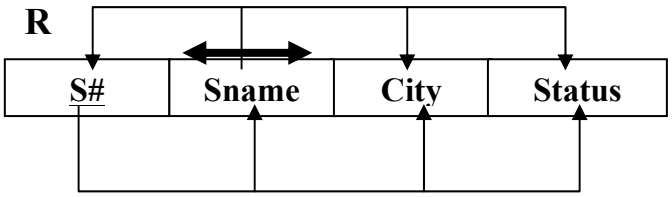


(4)نموذج تطبيع بويس كود(BCNF: Boyce-Codd Normal Form):

العلاقة R تكون على نموذج التطبيع بويس كود إذا:
1. تنطبق على 3NF.
2. إذا كانت المحددات الوحيدة هي المفاتيح المرشحة:

$\Rightarrow X \rightarrow Y$, X must be Candidate Key.

مثال:



: Candidate key or Super Key

هذا المثال ينطبق على BCNF.

التعديلات على 3NF,2NF من قبل بويس و كود:

قام العلمان بويس وكود بتطوير مفاهيم نماذج التطبيع الثاني و الثالث ليشمل مفهوم المفاتيح المرشحة وليس فقط مفهوم المفتاح الرئيسي، كالتالي:

(5) نموذج التطبيع الثاني(2NF) حسب بويس كود:

العلاقة تكون على نموذج التطبيع الثاني إذا كان:

1. يجب أن ينطبق على 1NF.
2. إذا كانت لكل خاصية غير مفتاحية (Non Prime) في العلاقة R يجب أن تكون تعتمد كلياً على كل المفاتيح المرشحة الموجودة في R.

مثال:

أذا كانت X, Y المفتاحين المرشحان، وكانت A أي حقل غير مفتاحي فأن:

$$\Rightarrow X \rightarrow A \;\&\; Y \rightarrow A$$

(6) نموذج التطبيع الثالث(3NF) حسب بويس كود:

العلاقة R تكون على نموذج التطبيع الثالث إذا:

1. كانت على نموذج التطبيع الثاني.
2. **إذا كان لكل الإعتمادات التطبيقية في العلاقة R:**

$$X \rightarrow A$$

إذا كانت X هي مفتاح فائق في R، فإن A هي خاصية مفتاحية (مفتاح رئيسي أو جزء مفتاح رئيسي).

(7) نموذج بويس كود المعدل:

العلاقة R تكون على نموذج التطبيع بويس كود إذا:

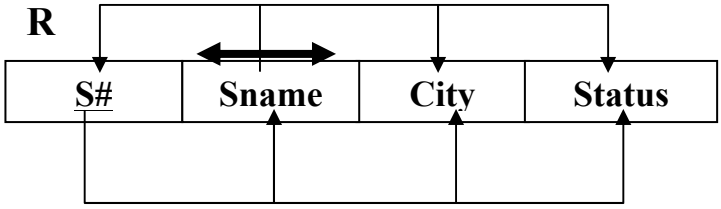
3. كانت على نموذج التطبيع الثالث.
4. **إذا كان لكل الإعتمادات التطبيقية في العلاقة R:**

$$X \rightarrow A$$

يجب أن تكون على الأقل مفتاح فائق(Super Key).

أسئلة الوحدة:

س[1]) هل العلاقة التالية تحقق نموذج بويس كود المعدل ؟ ولماذا؟



الوحدة الخامسة: لغة الاستعلام البنيوية (SQL)

الأهداف العامة للوحدة:

1. أن تعَرف الطالبة لغة الاستعلام البنيوية(SQL).
2. أن تميز الطالبة بين SQLولغات المستوى العالي الأخرى.
3. أن تحدد الطالبة العلاقة بين SQLوبين الجبر العلائقي.
4. أن تفرق بين جمل تعريف البيانات(DDL) وجمل معالجة البيانات(DML).
5. أن تكتب الطالبة استعلامات برمجية باستخدام الجمل التالية بشكل صحيح بنسبة 100%:
   - الاختيار SELECT
   - الإدخال Insert into.
   - التعديل و الحذف DELETE/UPDATE.
   - انشأ وتغيير الجداول CREATE/DROP/Alter
   - أنشا فهرس Create Index.
   - أنشا مظهر بيانات Create View.

مقدمة:

في هذه الوحدة نهدف إلى تعليمك أسس قواعد البيانات العلائقية Relational Data Bases وكيفية التحكم بها عن طريق لغة SQL البسيطة، تمكنك لغة SQL من إدارة قواعد البيانات بشكل كامل وإجراء جميع العمليات القياسية كإنشاء الجداول وتعبئتها بالبيانات، أو إجراء الاستعلامات عليها وكذلك الربط بين الجداول المختلفة.

ستتم كافة التمارين والأمثلة باستخدام قاعدة البيانات لشركة(company.mdb)، حيث أنها ستعطى لكل طالبة مع بداية التدريس لهذه الوحدة.

سنتبع طريقة التعلم باستخدام الأمثلة، لأنها الطريقة الأمثل لدراسة لغات البرمجة بالشكل الأوفى، و بسرعة.

قصة SQL:

تبدأ قصة SQL مع بداية النموذج العلائقي لقواعد البيانات من خلال ورقة بحثية أعدّها E.F.Codd سنة 1970 حين اشتغاله في مختبر أبحاث IBM في سان خوزيه . وفي سنة 1974 قام D. Chamberlin من نفس المختبر بتوصيف لغة سمّيت ' Structured English Language Query' ، أو SEQUEL . ثم أعدّت نسخة منقحة سنة 1976 عرفت باسم SEQUEL/2 ، ولكن سرعان ما تمّ تغيير الاسم الى SQL لأسباب قانونية (ذلك لأنه وجد ان نفس الاسم قد سبق تخصيصه واستخدامه من قبل آخرين) . وبرغم من أن العديد من الأفراد لا يزالوا ينطقون SQL بـ 'see-quel' ، إلا أن النطق الرسمي لها هو 's-q-l' .

بعد ذلك قامت IBM بإنتاج مجسّم لنظام إدارة قواعد بيانات DBMS أسمته System R ، أسس على لغة SEQUEL/2 ، وكان الغرض من هذا المجسّم هو اختبار مدى فعالية وجدوى النموذج العلائقي. فكان بجانب النجاحات التي تحققت فإن إحدى أهم النتائج التي رافقت هذا المشروع هو تطوير SQL .

عموما فإن جذور SQL تنبثق من لغة SQUARE أي (Specifying Queries as Expression Relational) ، والتي سبقت مشروع System R . حيث تمّ تصميم SQUARE لتكون لغة بحث تقوم بتنفيذ عمليات جبر علائقية بواسطة جمل إنكليزية.

في أواخر السبعينات قامت المؤسسة التي تعرف الآن باوراكل ORACLE Corporation بإنتاج نظام قاعدة بيانات ORACLE ، وكانت تقريبا أول منتج تجاري لنظام إدارة قواعد بيانات علائقي مبني على SQL . تبعه بعد ذلك بقليل INGRES مع لغة استفسار تسمى QUEL والتي برغم إنها كانت اكثر هيكلية من SQL إلا أنها اقل شبها بالإنكليزية. إلا أنه عندما ترسّخت لغة SQL كلغة معتمدة لقواعد البيانات تم تحويل INGRES إلى نظام إدارة قواعد بياناتDBMS مبني على SQL .

أما IBM فقد قامت بإنتاج أول نظام إدارة قواعد بيانات علائقي لها RDBMS أسمته SQL/DS لبيئة تشغيل DOS/VSE ، و لبيئة تشغيل VM/CMS وذلك في سنتي 1981 و 1982 على التوالي ، ثم فيما بعد لبيئة تشغيل MVS في 1983 وعرف باسم DB2 .

في عام 1982 بدأ المعهد القومي الأمريكي للمواصفات (ANSI) بالعمل على لغة قواعد بيانات علائقية (RDL) بالاعتماد على ورقة تصوّر قدّمتها IBM . ثم التحقت منظمة ISO لهذا العمل وقاما سويّا بتحديد مواصفة SQL . (تم إسقاط اسم RDL في 1984 ، و أعيد صياغة مسودة المواصفة بشكل يشبه التي عليه SQL الآن.)

و قد جوبهت المواصفة المبدئية من أيزو ISO التي نشرت سنة 1987 ، بالعديد من الانتقادات ؛ فقد صرّح "ديت" Date - وهو باحث له تأثيره في هذا المجال - بأن خصائصا مهمة مثل قواعد التكامل المرجعي وعمليات علائقية معينة قد تم شطبها من المواصفة، كما أشار إلى أن اللغة كانت تضج بالمرادفات ، أي وجود أكثر من طريقة لكتابة نفس الاستفسار . و قد كانت معظم الانتقادات صحيحة ، و تم أخذها في الاعتبار من قبل الجهات المختصة بالمواصفات قبل إصدار المواصفة . و قد تقرر إن الأمر الأكثر ضرورة هو إصدار مواصفة بأسرع وقت ممكن و ذلك

لإرساء قاعدة عامة تكون أساسا لتطوير اللغة ، بدلا من الانتظار حتى ذلك اليوم الذي يُمكن فيه الإجماع على كل الخصائص التي يشعر الناس بضرورة وجودها .

في 1992 أصدرت ايزو أول مراجعة مهمة لمواصفة SQL ويشار إليها عادة باسم SQL2 أو SQL-92 ، وبرغم من إن بعض الخصائص قد تم تحديدها لأول مرة في هذه المواصفة ؛ إلا أن معظمها قد سبق تنفيذها بشكل أو آخر في العديد من المنتجات.

و تبقى هناك بعض الخصائص والمزايا التي يضيفها مصنعو نظم SQL والتي تسمى ملحقات extensions ، هذه الملحقات تشكل لهجات dialect مختلفة تبتعد بمرور الوقت عن لغة المواصفة الأصلية. وبرغم ذلك يوجد الآن اتجاه قوي للعمل على توحيد مواصفات لغة SQL .

و الآن وبرغم إن SQL هو تصوّر أبدعته أصلا IBM ؛ فان أهمية هذه اللغة حثّت العديد من المصنعين الآخرين كي يقوموا بصنع إنتاجاتهم الخاصة ، فاصبح عدد المنتجات المتوفرة اليوم بالمئات .

ما هي لغة SQL ؟

- لغة الاستعلام البنيوية(Structured Query Language)
  - لغة برمجية خاصة تستخدم في:
    - بناء قواعد البيانات العلائقية(بناء جداول، حذف جداول، .....الخ)
    - معالجة البيانات في قواعد البيانات العلائقية (اختيار بيانات, إدخال بيانات...الخ)
    - تتميز SQL عن لغات المستوى العالي في أنها لغة خاصة بنوع معين من التطبيقات وهو قواعد البيانات العلائقية

العلاقة بين SQL و بين الجبر العلائقي

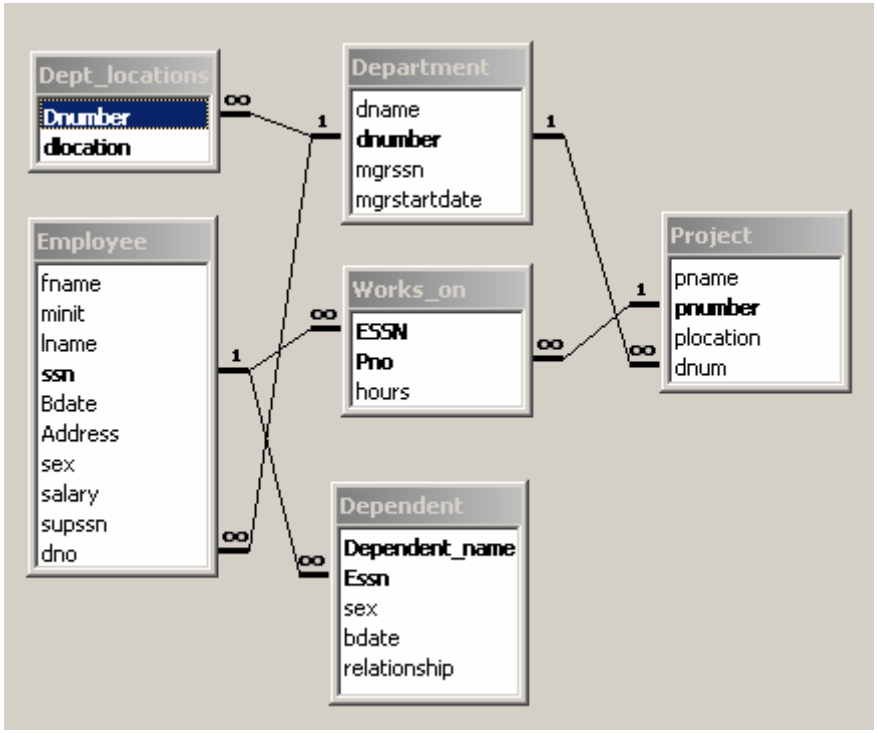
- الجبر العلائقي تمثل الأساس الرياضي للغة SQL
- SQL بنيت على أساس العمليات التي تطبق على المجموعات و العلاقات, ولكن مع بعض التغييرات و التحسينات
- سنأتي إلى ذكر العلاقة بينها عند شرح الجمل في SQL

جمل تعريف البيانات(DDL) و جمل معالجة البيانات(DML) في لغة SQL

- جمل تعريف البيانات(Data Definition Language)
  - تختص بتعريف الجداول و الحقول و العلاقات بين الجداول في قاعدة البيانات
  - أمثلة:CREATE, DROP, ALTER
- جمل معالجة البيانات(Data Manipulation Language)
  - تختص بالتعامل مع البيانات من اختيار، حذف، تعديل، أو اضافة
  - أمثلة:SELECT, INSERT INTO, DELETE, UPDATE

قاعدة البيانات لشركة(Company Database):

- ستكون هذه القاعدة للبيانات هي أساس التمارين و الأمثلة في هذه الوحدة، نظرا لكونها تمكننا من تطبيق كافة أنواع الاستعلامات عليها.
- و إليكم النموذج العلائقي لها:



شكل 1: قاعدة بيانات لشركة

- سيحصل كل طالب على نسخة من قاعدة البيانات، على شكل ملف قاعدة بيانات لنظام إدارة قواعد البيانات (Microsoft Access)، بالاسم (Company.mdb).
- للحصول على معلومات حول كيفية كتابة الاستعلامات في Access يرجى الرجوع إلى دوسية Microsoft access، المعطية من قبل أستاذ المساق، ودراسة وحدة أنشاء الاستعلامات

جملة الاختيار (SELECT)

- الشكل العام لها:

select $A_1, A_2, \ldots, A_n$
from $r_1, r_2, \ldots, r_m$
where P

A_is represent attributes
r_is represent relations
P is a predicate.

- تماثل في الجبر العلائقي:

$$\Pi_{A_1, A_2, \ldots, A_n}(\sigma_P(r_1 \times r_2 \times \ldots \times r_m))$$

- الناتج لعملية الاختيار هو علاقة(جدول)

أقسام جملة SELECT الرئيسية

1. **Select clause**
2. **From clause**
3. **Where clause**

- **سيتم الحديث عن كل قسم من هذا الأقسام بإسهاب في الصفحات التالية.**
- **هنالك أقسام أخرى سنتحدث عنها لاحقاً.**

1. قسم الاختيار(Select clause)

- تماثل عملية الإسقاط(project) في الجبر العلائقي
  - مثال:

```
Select fname
From Employee
```

تمثل

$$\Pi_{fname}(Employee)$$

- وجود * تعني اختيار كل الحقول من الجدول المطلوب مثال:

```
Select *
From Employee
```

- لغة SQL تسمح بالتكرارات في الجداول وفي النتائج للاستعلام
- لمنع التكرارات من الحصول نضيف كلمة(distinct) بجانب كلمة Select

Select distinct salary From Employee;	تعني: استخرج رواتب جميع الموظفين بدون تكرار للرواتب المتساوية

- الكلمة all تعمل عكس عمل distinct

Select all salary From Employee;	تعني: استخرج رواتب جميع الموظفين مع التكرار للرواتب المتساوية أن وجد

- كلمة select قد يأتي بعدها عمليات حسابية على الحقول (+,*,-, /)

❑ مثال:

تعني : استخرج أرقام و رواتب جميع الموظفين مع زيادة مقدارها 10%	Select ssn, salary*1.1 From Employee;

2. قسم الشرط(Where clause)

■ ما يأتي بعد كلمة where يمثل الشرط على استخراج البيانات

❑ مثال:

تعني : استخرج أرقام و رواتب جميع الموظفين الذكور	Select ssn, salary From Employee Where sex="M";

■ يمكن استخدام أدوات الربط المنطقية(And, Or, not) لربط مجموعة من الشروط

تعني : استخرج أرقام و رواتب جميع الموظفين الذكور ورواتبهم أكبر من 25000	Select ssn, salary From Employee Where sex="M" and salary>25000;

■ يمكن استخدام كلمة (Between….and….) لتبسيط الشرط

❑ مثال:

تعني : استخرج أرقام جميع الموظفين و لهم راتب أكبر من 25000 و اقل من 40000	Select ssn From Employee Where salary between 25000 and 40000;

3. قسم مصدر البيانات(from clause)

■ تمثل عملية الضرب الكارتيزي في الجبر العلائقي

- مثال:

Select ssn, dnumber From Employee, department;	النتاج يكون جدول يتكون من مجموع حقول الجدولين(Employee, Department)

- تمثل عملية الدمج(Join) في الجبر العلائقي إذا كان الشرط يحتوي على تساوي للمفتاح الرئيسي والمفتاح الغريب للجدولين.

```
Select ssn, dnumber
From Employee, department
Where Employee.dno = Department.dnumber;
```

- يمكن تمثيل عملية الدمج(Join) كما في الجبر العلائقي بطريقة أخرى، باستخدام أما:
  - Inner Join
  - Left Outer Join
  - Right Outer Join

  بالإضافة إلى شرط الدمج في جزء مصدر البيانات.
  - مثال:

```
Select ssn, dnumber
From Employee Inner Join department on Dno=Dnumber;
```

- إعادة التسمية لحقل
  - مثال:

```
Select ssn, salary*1.1 As newsalary
From Employee;
```

- إعادة التسمية للجدول
  - مثال:

```
Select ssn, dnumber
From Employee as E, department as D
Where E.dno = D.dnumber;
```

معالجة السلاسل الرمزية (String Operations)

- لغة SQL تحتوي حروف بديلة (Wild-Cards) يمكن استخدامها في عمليات المقارنة على السلاسل الرمزية، وهذه الحروف البديلة هي:
  - حروف تحل محل عدد من الحروف غير معروف، ونستخدم الرمز (*) لهذا الغرض.
  - حروف تحل محل حرف واحد غير معروف، ونستخدم الرمز (?) لهذا الغرض.
  - نستخدم أدوات التنصيص (" ")، للإحاطة بالسلاسل الرمزية، نستخدم الرمز (# #) للإحاطة بالتواريخ و الوقت.
- وكما نلاحظ فان هذه الحروف تشبه في عملها، عملية البحث عن الملفات في Windows.
- يجب استخدام كلمة LIKE في الشرط إذا استخدمت الحروف البديلة
  - أمثلة:

Select fname From Employee Where Address like "*Houston*";	تعني: استخرج اسماء الموظفين الذين عنوانهم يحتوي كلمة Houston.
Select fname From Employee Where fname like "?a???";	تعني: استخرج اسماء الموظفين، الذين اسماءهم تتكون من 5 أحرف ، و الحرف الثاني منها هو حرف a .

Select fname From Employee Where bdate Between #1/1/70# and #1/1/90#:	تعني: استخرج أسماء الموظفين، الذين تواريخ ميلادهم تتراوح بين عام 70 و 90.
Select fname From Employee Where fname like "[A-M]*";	تعني: استخرج أسماء الموظفين الذين أسمائهم تبدأ بإحدى الحروف التالية: A, B, C… M

ترتيب النتائج (Ordering Results)

- نستخدم الجملة البرمجية (Order By)، من أجل ترتيب العناصر الناتجة من الاستعلام

ويجب أن تتبع هذه الجمل بـِ:

- حقل الترتيب(Ordering Field).
- نوع الترتيب، تصاعدي(asc) أم تنازلي (desc).
- أمثلة:

Select fname, Lname From Employee Order by fname asc;	تعني: استخرج الأسماء الأولى و الأخيرة للموظفين، ورتب الناتج بحسب الاسم الأول تصاعدياً.
Select fname, Lname From Employee Order by Lname desc;	تعني: استخرج الأسماء الأولى و الأخيرة للموظفين، ورتب الناتج بحسب الاسم الأخير تنازلياً.
Select fname, Lname, salary From Employee Order by salary asc, fname desc;	تعني: استخرج الأسماء الأولى و الأخيرة و الرواتب للموظفين، ورتب الناتج بحسب الراتب تصاعدياً، و إذا تساوى الراتب، رتبه بحسب الاسم الأول تنازلياً

العمليات الحسابية (Aggregate Functions)

- هذه العمليات تطبق على مجموعة من القيم لحقل في علاقة معينة ، و تُرجع قيمة واحدة.

- وهذه العمليات هي:

1. **avg( )** : Average value.
2. **min**( ) : minimum value.
3. **max**( ) : maximum value.
4. **sum**( ) : sum of values.
5. **count( )**: number of values.

- عند استخدام هذه العمليات مع حقول عادية، يجب استخدام الجملة البرمجية التجميع بحسب (group by)، ويكون معامل group by هو تلك الحقول العادية.
- الشرط على group by يكون باستخدام جملة (Having)، وتأخذ نفس الشروط لجمل where.

❑ أمثلة:

Select max(salary), min(salary),avg(salary) From Employee	تعني: أستخرج أعلى و اقل راتب، و معدل الرواتب من جدول الموظفين.
Select count (*) From Employee	تعني: أستخرج عدد السجلات في جدول الموظفين. **مع ملاحظة أن count تعدد السطور وتعدد من ضمنها القيم الفارغة(NULL)**
Select fname, lname From Employee Where (select count (*) from dependent Where ssn=Essn)>=2;	تعني: استخرج الاسم الأول و الأخير لكل الموظفين الذين لهم معالين أو أكثر. **مع ملاحظة استخدام فكرة الاستعلام المتضمن(Nested Query).**
Select dno, count (*), avg(salary) From Employee Where dno <> NULL Group by Dno;	تعني: لكل دائرة، استخرج رقم الدائرة، وعدد الموظفين فيها، ومعدل رواتب موظفيها. **مع ملاحظة استخدام أداة المقارنة <> و تعني لا يساوي.**
Select pnumber, pname, count (*) From works_on, project Where pno=pnumber Group by pnumber, pname Having count (*)>2;	تعني: لكل مشروع يعمل به أكثر من موظفين، أستخرج اسم المشروع و رقمه، وعدد الموظفين الذين يعملون به.

Select dname, count (*) From Employee, Department Where Dnumber=Dno and salary>40,000 and dno IN (Select Dno from Employee Group by dno Having count (*)>5) Group by dname;	تعني: لكل دائرة فيها أكثر من خمسة موظفين، أستخرج اسم الدائرة، و عدد الموظفين الذين يحصلون على رواتب تزيد عن 40, 000 **مع ملاحظة استخدام كلمة IN و التي تعني هل أن Dno أحدى النتائج للاستعلام المتضمن.**

جمل التغيير على البيانات (Date changing)

- وتشمل الجمل التالية:

1. Insert into
2. update
3. delete

- وسنلاحظ من خلال الأمثلة عليها مدي التشابه بينها و بين جملة الاختيار(select)

إضافة سجلات(Insert Into)

- وتستخدم من اجل إضافة سجلات جديدة إلى جدول معين:
  - أمثلة:

Insert into department Values (6,'R&D','999998544','1/1/2002')	تعني: ادخل السجل الموجود بعد كلمة values إلى جدول الدائرة، بنفس ترتيب الحقول الموجود في التصميم للجدول
Insert into department (dname, dnmuber, mgrssn, mgrstartdate) Values ('R&D', 6,'999998544','1/1/2002')	تعني: ادخل السجل الموجود بعد كلمة values إلى جدول الدائرة، بنفس ترتيب الحقول الموجود في الاستعلام.

حذف سجلات(delete)

- وتستخدم من اجل حذف سجلات من جدول معين:

- يجب أن تستخدم بحذر شديد لأن استخدامها بشكل خاطئ، قد يؤدي إلى فقدان البيانات.
  - أمثلة:

تعني: احذف سجل الموظف الذي رقمه 123456789	Delete from Employee Where ssn="123456789";
تعني: أحذف سجلات الموظفين الذين يعملون في الدائرة التي تسمى Research.	Delete from Employee Where Dno in (Select dnumber from department where dname="Research");

تعديل السجلات(Update)

- وتستخدم من اجل تعديل بيانات معينة في سجلات من جدول معين:
  - أمثلة:

تعني: أضف زيادة بمقدار10% على رواتب الموظفين الذين يعملون في الدائرة رقم 5.	Update Employee Set salary=salary*1.1 Where dno=5;

جمل تعريف البيانات (DDL)

- ملاحظة: هذه الجمل غير مدعومة في Access ، وتستخدم في قواعد البيانات الكبيرة مثل أوراكل، SQL Server، وغيرها.

جملة (Create)

- وتأتي على ثلاثة أشكال:

1. Create table
2. Create index
3. Create view

جملة انشأ الجداول (Create Table)

- وتستخدم لتعريف جدول، مع تعريف أنواع البيانات، و الضوابط على الحقول، والعلاقات بين الجداول.
  - مثال بأستخدام SQL+ لاوراكل:

تعني: ابني جدول الموظفين	Create Table Employee (Fname varchar(20) not NULL, SSN char (9) not NULL default "111111111", Bdate Date, Salary Decimal (10,2), Dno Int default 1, Supssn char (9), Primary key (SSN), Unique (Fname), Foreign key (Dno) references department (Dnumber) on Delete set NULL on update cascade, Foreign key (Supssn) references employee (SSN) on Delete set NULL on update cascade,);

جملة حذف الجداول (Drop Table)

تعني: أحذف الجدول Dept_Loc و وزع التأثيرات من الحذف على الجداول المرتبطة	Drop Table Dept_Loc cascade;

جملة تغيير التصميم لجداول (Alter Table)

Alter Table Employee Add jobtitle char(10);	تعني: أضف الحقل jobtitle الى جدول الموظفين.
Alter Table Employee drop bdate;	تعني: أحذف الحقل bdate من جدول الموظفين.

جملة انشأ فهرس بيانات (Create Index)

Create index EmpIDX on Emp(fname asc, bdate)	تعني: أنشاء فهرس باسم EmpIDX على الحقول fname و bdate ، مع ترتيب الفهرس بحسب الأسم الاول في جدول الموظفين.

جملة انشأ مظهر بيانات (Create View)

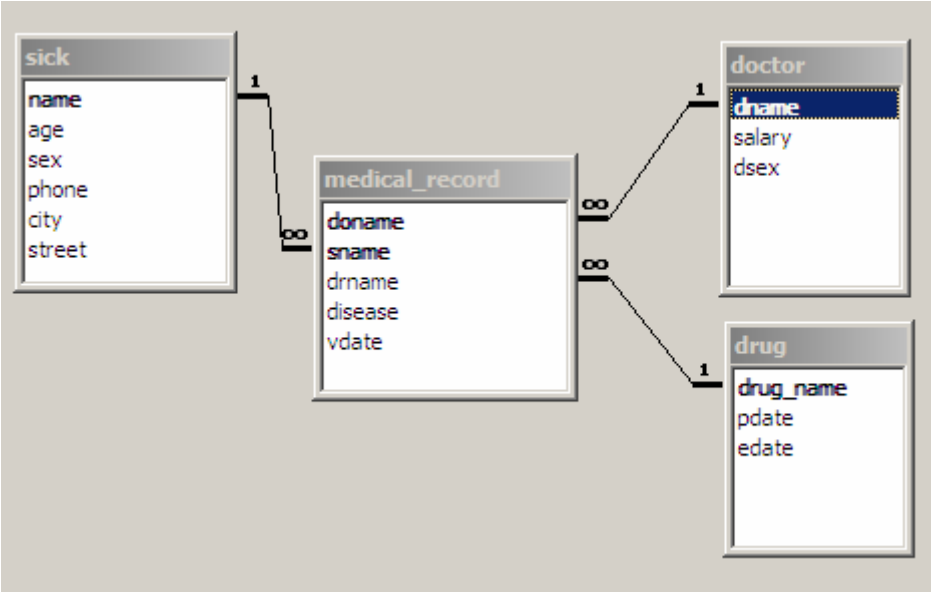
- وتستخدم لتعريف جدول وهمي(Virtual Table).

Create view EmpFullName on select fname, minit, lname from Employee;	تعني: أنشاء مظهر بيانات من جدول الموظفين يحتوي على الحقول: fname و Minit و lname. وسميه EmpFullName

أسئلة الوحدة:

1. قارن من خلال جدول بين جملة SELECT في الجبر العلائقي و في SQL ؟
2. وضح الفرق بين جمل DDL و DML في SQL من خلال ثلاثة نقاط ؟

بالأخذ بعين الاعتبار الجداول التالية التي تمثل قاعدة بيانات لمستشفى(Hospital):



شكل 2: قاعدة بيانات لمستشفى

أجيب على الأسئلة التالية باستخدام لغة الاستعلام البنيوية (SQL):

3. صمم استعلام تحديد يقوم بعرض أسماء المرضى بحسب الجنس.
4. صمم استعلام يعرض سجلات الأطباء ذوي الرواتب التي تتراوح بين 2000 و 3000.
5. صمم استعلام بكل المرضى المصابين بمرض القلب(Heart) مع عرض أسم الطبيب المعالج وكل أدوية العلاج.
6. صمم استعلام يعرض قائمة بسجلات الأدوية التالفة(التي انتهى تاريخها).
7. اكتب الأجراء اللازم لبناء جدول (medical_record).
8. استخرج عدد المرضى المصابين بمرض السكري(sugary).

المراجع

1. **قاعدة البيانات و أدارتها**، جامعة القدس المفتوحة، الطبعة الأولى 1997.

2. الطيطي، جعفر 2001, **Microsoft Access**, مدرس حاسوب تقني، خطة ثنائية، كلية العلوم التربوية وكلية مجتمع المرآة.

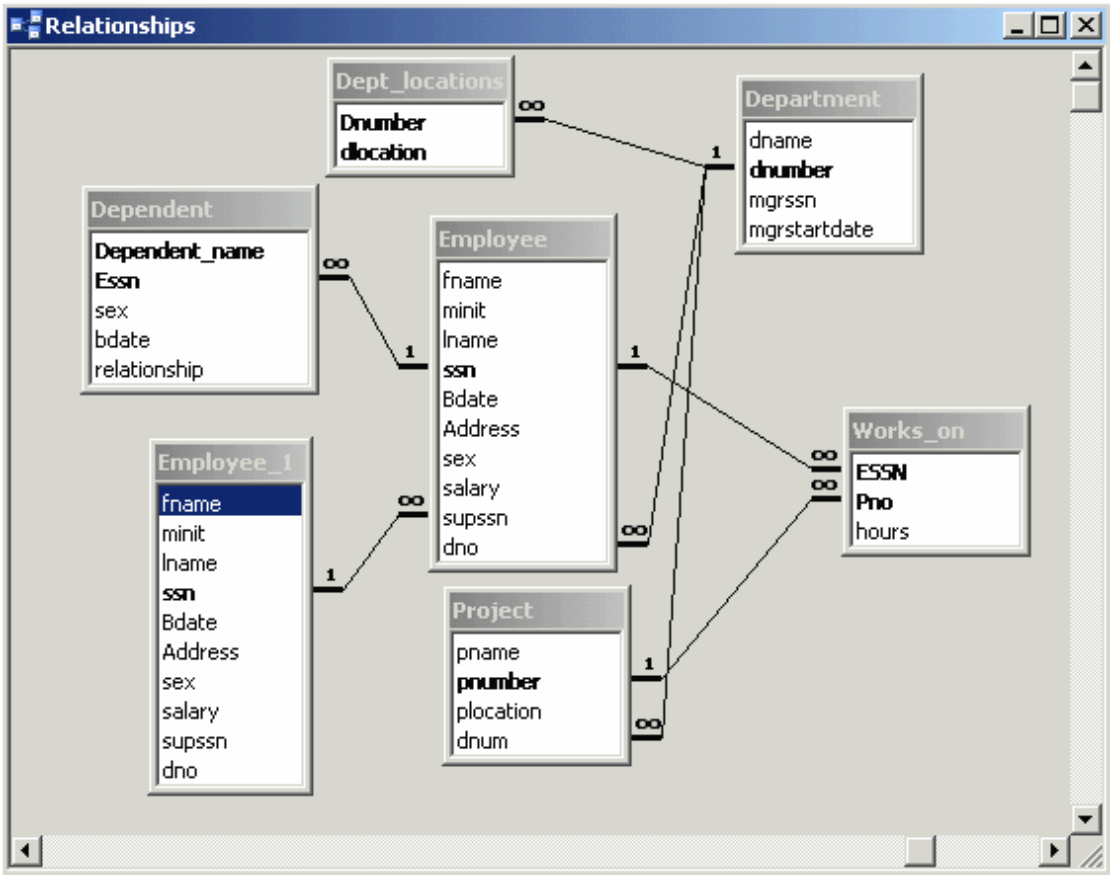
3. الطيطي، جعفر, **أثر تطوير وحدة البرمجة بلغة الاستعلام البنيوية(SQL) على تحصيل الطالبات ضمن مساق معالجة قواعد البيانات**, مدرس حاسوب تقني، بحث أجرائي ضمن دورة التربية في أثناء الخدمة التي يعقدها معهد التربية – الأنروا، كلية العلوم التربوية وكلية مجتمع المرآة، **تموز 2002.**

4. **نظم إدارة قواعد البيانات و تطبيقاتها**، موسوعة دلتا كمبيوتر.

5. Elmsari & Navathe (2000), **Fundamentals of Databases Systems**, Third Edition.

6. Silberschatz, Korth, (1997), **Database Systems Concepts**.

ملحق: قاعدة بيانات لشركة



Employee : Table

رقم الدائرة	رقم المشرف	الراتب	الجنس	العنوان	تاريخ الميلاد	رقم الموظف	الاسم الاخير للموظف	الاسم الثاني للموظف	الاسم الاول للموظف
4	987654321	43000	M	654 Voss,Houston,TX	12-Apr-78	034863217	Hamidah	K	Mohammad
5	333445555	30000	M	731 Fondren, Houston,TX	09-Jan-55	123456789	Smith	B	Jone
5	888665555	40000	M	638 voss, houston, TX	08-Dec-45	333445555	Wong	T	Franklin
5	333445555	25000	F	5631 Rice,Houston,TX	31-Jul-62	453453453	English	A	Jocy
5	333445555	50000	M	543 Rice,Houston,TX	09-Mar-62	544332115	Abed	M	Mustafa
5	333445555	38000	M	975 Fire Oka, Humble, TX	15-Sep-52	666884444	Narvan	k	Ramesh
1		55000	M	450 Stone,houston,TX	10-Nov-27	888665555	Borg	E	James
4	034863217	25000	M	756 Berry,Bellaire,TX	08-Aug-40	902828896	Ali	E	Khaled
1	987654321	40000	M	732 Fondren,Houston,TX	09-Jul-56	954231285	Ahmad	H	Jamal
1	888665555	45000	M	984 Dallas,Houson,TX	04-Dec-59	973651306	Salah	A	Sameer
4	888665555	43000	F	291 berry, bellaire, TX	20-Jun-31	987654321	Wallace	S	Jennifer
4	987654321	25000	M	980 Dallas,Houston,TX	29-Mar-59	987987987	Jabber	V	Ahmad
4	987654321	25000	F	3321 castle, sonng, TX	19-Jul-58	999887777	Zelaya	J	Alica
0		0				111111111			

Record: of 13

Department : Table

اسم الدائرة	رقم الدائرة	رقم المسؤول	تاريخ بدء عمل المسؤول
Headquarters	1	988665555	19-Jun-71
Administration	4	987654321	01-Jan-85
Research	5	333445555	22-May-78

Record: 1 of 3

Dependent : Table

اسم المعال	رقم الموظف	الجنس	تاريخ الميلاد	صلة القرابة
Abner	987654321	M	29-Feb-32	SPOUSE
Alica	123456789	F	30-Sep-78	DAUGHTER
Alice	333445555	F	05-Apr-76	DAUGHTER
Elizabeth	123456789	F	05-May-57	SPOUSE
Joy	333445555	F	03-May-48	SPOUSE
Michael	123456789	M	01-Jan-78	SON
Theodore	333445555	M	25-Oct-73	SON

Record: 1 of 7

Project : Table

اسم المشروع	رقم المشروع	الموقع	رقم الدائرة
ProductX	1	Bellatre	5
Computerizatior	10	Stafford	4
ProductY	2	Sugarland	5
Reorganization	20	Houston	1
ProductZ	3	Houston	5
Newbenelits	30	Stafford	4
			0

Record: 6 of 6

Dept_locations : Table

رقم الدائرة	الموقع
1	Houston
1	Humble
4	Sonng
4	Stafford
5	Bellaire
5	Houston
5	Sugerland
0	

Record: 1

Works_on : Table

رقم الموظف	رقم المشروع	عدد الساعات
034863217	2	40
034863217	3	10
123456789	1	32
123456789	2	8
333445555	10	10
333445555	2	10
333445555	20	10
333445555	3	10
453453453	1	20
453453453	2	20
544332115	20	33
666884444	3	40
902828896	10	40
954231285	10	5
973651306	1	8
973651306	20	15
999887777	30	30
		0

Record: 2 of 17

Jafar Ishaq Titi

جعفر اسحق طيطي

RWTC, Al-Tereh Street, Ramallah, P.O.Box 214
jafart@gmail.com

Profile

M.S. degree in Scientific Computing. B.S. degree majoring in Electrical Engineering and minor in Computer Science, in addition holds a diploma in education. Also a Skilled research engineer and programmer, Experienced computer technical instructor. And an Expert web designer.

Education

M.S. Scientific Computing, Birzeit University/ Palestine — *February 2005*

B.S. Electrical Engineering, Birzeit University/ Palestine
Major: Electrical Engineering(Light Current)
Minor: Computer Science — *September 2000*

Diploma in education, Education Faculty/UNRWA
Vocational training instruction — *August 2003*

Publications

Master's of Science Thesis
Jafar I. Titi, *Simulation, Analysis, and Advanced Control of Selected Nonlinear Dynamic Systems*, December 2004.

Invited Paper/Presentation
Jafar I. Titi, Haithem Abu-Rub, "Comparative Analysis of Fuzzy and PI Sensorless Control Systems of Induction Motor Using Power Measurement", CPE'05 conference, Gdynia, Poland, June 2005.

Books and Course Materials
Jafar I. Titi, *Fundamentals of Database Systems*, RWTC, UNRWA, June 2003.

Career History

Educational sciences Faculty and Ramallah women Training Center, UNRWA, *Ramallah, Palestine(http://www.rwtc.edu)* — *Sept 2000-current*

Computer Technical Instructor

- Teaching courses of computer science; Cisco Networking CCNA, Database, Data Structure, Software Engineering, Web Design, Macromedia Flash and other Desktop Application Courses.
- Supervises several innovative Graduation Projects for the computer department students.
- Helping revise curriculum and graduation requirements for students in computer section to better prepare them for the job market.
- Teaching Courses of computer science for ESF Students.
- Teaching Community short-term courses in computer field.
- RWTC/ESF Website Designer and master.

Al-Quds Open University, *Ramallah, Palestine (http://www.qou.edu)* — *Oct. 2005-current*

Computer Academic Lecturer, Communications and IT dept.

Saffa club, Saffa, *Ramallah, Palestine* — Aug 2005-Jan 2006

Teahing a 70 hour web Design Course includes: Process And Specifications for Web Design, html, FrontPage, Flash, image editing, SQL server, asp.

ClaraNET ISP, *London, UK (http://www.claranet.com)* — *1999*

Web Designer

- UNIX (FreeBSD), Apache Web Server, Bind DNS server, MTA Mail Server, and X-Windows server administration.
- Internet project that includes web interface for a Database, MySQL Database server, Perl language, Html.

Computer Center, *Birzeit University, Palestine* *March 1998- March 1999*

Support Team Member

- Network connections (hardware & software).
- Installing various computer hardware & software.
- Computer hardware & software maintenance.

Awards

- Dean's Honor List for 6 semesters, Msc. Scientific Computing, Birzeit.
- Dean's Honor List for two semesters, Electrical Engineering, Birzeit University.
- Summer training grant: ClaraNET ISP, London, UK

Professional Memberships

- Palestine and Jordan Engineers Associations, Electrical Engineering Div. *Sept. 2000*
- Certified Computer Network Associate (CCNA) Cisco Academy Instructor *July 2002*
- International Computing Driving License (ICDL) Certified Instructor. *May 2003*

Skills and Interests

- **Programming Languages:** C++, Pascal, VB, FORTRAN, SQL, Assembly, and Matlab.
- **Web design and HTML programming:** Java and VB (applet's and scripts), working with Visual Editors of HTML, ASP, Flash animations.
- **Operating systems:** DOS, Win3.x/95/98/ME/XP/WindowsNT, and UNIX (FreeBSD, Linux).
- **Applications:** MS Office, AutoCAD, MathCAD, and Photoshop, Macromedia flash, Visio Technical, and others.

Projects and Seminars

- **Presentation and defense for the Master Thesis:** A novel and inexpensive method to develop and assess sensorless control schemes of induction motors. The proposed systems operate without speed sensor; the rotor angular speed is calculated in different ways. The methods are applied to field oriented control principle. Fuzzy logic controller used for speed control. A comprehensive analysis and comparative study of several relationships of rotor angular speed presented.
- **Several computer simulation programs in the field of scientific computing.**
- **Designer** and **webmaster** for a complete Website for RWTC.
- **B.Sc. Graduation Project**: Design of a complete Barcode generation machine: - Design of a computer controlled automated production line, includes printing labels, design of an input-output computer card, and database programming.
- Programming of a portion of an operating system (Virtual Memory Management): Object-Oriented C++.
- A **database program** for the registration office in Birzeit University Using Ms Access.
- Design of several computer systems using OrCAD. Suggested improvements on the CSMA/CD Protocol, Ethernet collision problems. Ethernet Technology: a complete overview, and future. Design of an Automatic Controller of water level in a tank, Types of memory, digital families.

Languages

Arabic (Native), **English** (Excellent), and **Hebrew** (Fair).