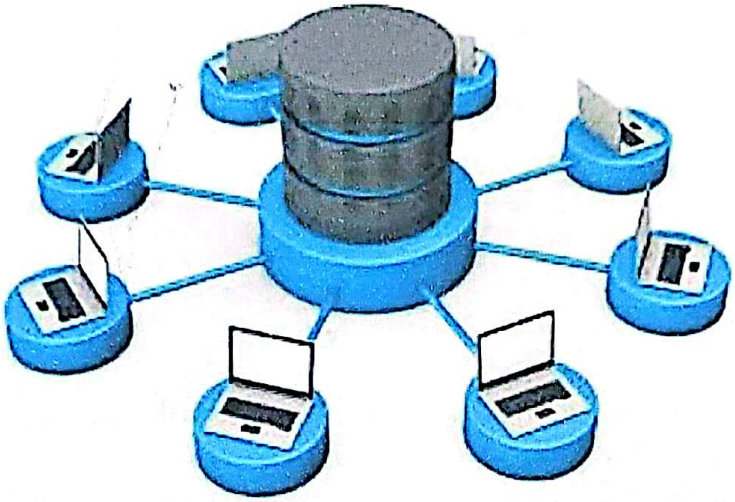
جامعة شط العرب الاهلية



Database Systems

المرحلة الثانية

الدراسة الصباحية / الدراسة المسائية

د. مها الغالبي

2025-2024

نظام بولونيا

محاضرة 1: مقدمة عن قواعد البيانات (Introduction to Databases)

1. الفرق بين البيانات والمعلومات:

- **البيانات (Data):** هي حقائق أو أرقام خام غير معالجة.
  مثال: درجات الطلاب قبل الحساب.
- **المعلومات (Information):** هي البيانات التي تم معالجتها لتصبح ذات معنى.
  مثال: متوسط درجات الطلاب.

2. تعريف قاعدة البيانات وقاعدة البيانات العلائقية:

- **قاعدة البيانات (Database):** مجموعة منظمة من البيانات التي يتم تخزينها إلكترونيًا.
  مثال: قاعدة بيانات تحتوي على معلومات الموظفين.
- **قاعدة البيانات العلائقية (Relational Database):** نوع من قواعد البيانات يعتمد على تنظيم البيانات في جداول مترابطة.
  مثال: جدول الموظفين وجدول الأقسام.

3. أنظمة إدارة قواعد البيانات (DBMS):

- **المفهوم:** برامج تُستخدم لإدارة قواعد البيانات بفعالية.
  مثال: MySQL، Oracle.
- **الأهمية:** تسهل تخزين البيانات واسترجاعها بطريقة منظمة وآمنة.

محاضرة 2: خصائص ومزايا وعيوب قواعد البيانات (Characteristics, Advantages, and Disadvantages of Databases)

1. خصائص قواعد البيانات:

- **التنظيم** :ترتيب البيانات بطريقة منطقية لتسهيل استخدامها.
- **التكاملية** :استخدام قواعد قيود مثل المفاتيح الأساسية والأجنبية لضمان صحة البيانات.
- **القابلية للتوسع** :دعم أنظمة قواعد البيانات للتوسع المستقبلي.
- **التشارك** :إتاحة البيانات لأكثر من مستخدم مع الحفاظ على الأمن.

2. المزايا:

- **التكاملية** :تقليل التعارض بين البيانات من خلال الدمج.
- **الأمن** :توفير كلمات مرور وتشفير لحماية البيانات.
- **تقليل التكرار** :إزالة التكرار غير الضروري في البيانات.
- **استرجاع البيانات الفعال** :استخدام استعلامات مخصصة للوصول إلى المعلومات بسرعة.

3. العيوب:

- **التكلفة العالية** :تشمل تكلفة الأجهزة، البرمجيات، وتدريب الموظفين.
- **تعقيد النظام** :صعوبة تعلم الأنظمة المتقدمة.
- **مشاكل الأداء** :الحاجة إلى تحسينات مستمرة للحفاظ على الأداء مع نمو البيانات.
- **الاعتماد الكبير** :قد يؤدي تعطل النظام إلى توقف العمل.

المحاضرة 3: مراحل تصميم قواعد البيانات (Database Design Phases)

نظرة عامة على مراحل تصميم قواعد البيانات:

- تتضمن عملية تصميم قواعد البيانات عدة مراحل رئيسية تهدف إلى إنشاء نظام فعّال لإدارة البيانات.
- المراحل تشمل جمع المتطلبات، التحليل، التصميم والتنفيذ.

المرحلة الأولى: جمع المتطلبات:(Requirement Collection)

- هذه المرحلة هي الأساس لتصميم قاعدة البيانات حيث يتم جمع كل التفاصيل المطلوبة لتطوير النظام.

وتتم عن طريق:

- مقابلات مع المستخدمين النهائيين.
- استخدام استبيانات للحصول على مدخلات إضافية.
- دراسة الأنظمة الحالية وتحليلها.

- **الناتج:**
  - قائمة مفصلة بالمتطلبات الوظيفية وغير الوظيفية.
  - تحديد نطاق العمل.

المرحلة الثانية: تحليل المتطلبات:(Requirement Analysis)

- في هذه المرحلة، يتم تحليل المتطلبات المجمعة وتحديد العلاقات بين البيانات.

وتتم عن طريق:

- الرسوم البيانية لتدفق البيانات.(DFD)
- الجداول والرسوم التخطيطية لتصنيف البيانات.

- **الناتج:**
  - نماذج مبدئية للنظام.
  - مخطط الكيانات والعلاقات.(Entity-Relationship Diagram - ERD)

المحاضرة 4: مراحل تصميم قواعد البيانات

المرحلة الثالثة: تصميم النموذج المفاهيمي باستخدامERD

- **ERD (Entity-Relationship Diagram)**

هو مخطط يُستخدم لتمثيل الكيانات (Entities) والعلاقات (Relationships) بين هذه الكيانات في النظام.

- يهدف إلى تصور البيانات في شكل رسومي يساعد في الفهم والتواصل بين الفريق.

- **المكونات الأساسية لـ:ERD**
  - **الكيانات :(Entities)** تمثل الأشياء التي يتم تخزين المعلومات عنها مثل "الطلاب" أو "المواد."
  - **السمات :(Attributes)** تمثل خصائص الكيانات مثل "اسم الطالب" أو "تاريخ الميلاد."
  - **العلاقات :(Relationships)** تمثل كيفية ارتباط الكيانات ببعضها البعض مثل "الطالب يسجل في مادة."
- **الارتباطات :(Cardinality)** تحديد عدد العلاقات بين الكيانات مثل (واحد لواحد، واحد لكثير، كثير لكثير.
  - **One to One**
  - **One to Many**
  - **Many to many**
- **(كيفية رسم:ERD)**

1. تحديد الكيانات.
2. تحديد السمات لكل كيان.
3. تحديد العلاقات بين الكيانات.
4. إضافة القيود مثل.Cardinality

المرحلة الرابعة: تصميم النموذج المنطقي

- **التعريف بالنموذج المنطقي:**
  - النموذج المنطقي هو تمثيل مفصل للبيانات من خلال تحويل النموذج المفاهيمي (ERD)إلى هيكل قاعدة بيانات منطقي يمكن تنفيذه في النظام.
  - يتم التركيز على **الجدول(Tables) ، المفاتيح(Keys) ، و العلاقات بين الجداول.**
- **الخطوات الأساسية لتحويل ERD إلى نموذج منطقي:**
  - تحويل الكيانات إلى جداول.
  - تحويل السمات إلى أعمدة في الجداول.
  - تحديد المفاتيح الرئيسية (Primary Keys) لكل جدول.
  - تحديد المفاتيح الأجنبية (Foreign Keys) لربط الجداول معاً.

- تعريف القيود (Constraints) مثل القيود على القيم (Check Constraints).

- **الفرق بين النموذج المفاهيمي والنموذج المنطقي:**
  - النموذج المفاهيمي يركز على تمثيل البيانات بشكل مجرد في صورة علاقات، بينما النموذج المنطقي يركز على تحديد كيفية تنظيم هذه البيانات في هيكل قاعدة بيانات قابل للتنفيذ.

المحاضرة 6 – 5 : تمثيل الرموز في ERD

رموز الكيانات والعلاقات

1. رموز الكيانات:(Entities)
   - الكيانات تمثل العناصر أو الأشياء في النظام.
   - يتم تمثيل الكيانات في ERD عادةً على شكل مستطيل.

2. رموز العلاقات:(Relationships)
   - العلاقات تمثل كيفية تفاعل الكيانات مع بعضها البعض.
   - يتم تمثيل العلاقات في ERD على شكل ماسة أو معينية.

3. السمات: Attributes

وهي خصائص الكيانات ويتم تمثيلها بشكل بيضوي

تعريف الارتباط (Cardinality)

1. الارتباط :(Cardinality)
   - يحدد عدد المرات التي يمكن أن تحدث فيها العلاقة بين الكيانات.
   - الأنواع الرئيسية للارتباط هي:
     - واحد لواحد(One-to-One)
     - واحد لكثير(One-to-Many)
     - كثير لكثير(Many-to-Many)

المحاضرة 7: تحديد المفتاح الاساسي والمفتاح الاجنبي Primary Key و Foreign Key

تُعد المفاتيح في قواعد البيانات من العناصر الأساسية التي تساهم في تنظيم البيانات وضمان تكاملها وسلامتها. من بين هذه المفاتيح، نجد المفتاح الأساسي (Primary Key) والمفتاح الأجنبي (Foreign Key)، وهما يلعبان دورًا مهمًا في ربط الجداول ببعضها البعض وتحقيق التكامل.

أولًا: المفتاح الأساسي(Primary Key)

تعريف المفتاح الأساسي:

المفتاح الأساسي هو عمود أو مجموعة أعمدة داخل الجدول يتم استخدامه لتحديد كل سجل (صف) داخل الجدول بشكل فريد. لا يمكن أن يحتوي المفتاح الأساسي على قيم مكررة أو فارغة(NULL)

خصائص المفتاح الأساسي:

1. **يجب أن يكون فريدًا** :لا يمكن أن تتكرر قيم المفتاح الأساسي في أكثر من سجل داخل الجدول.
2. **لا يقبل القيم الفارغة** :(NULL) يجب أن يحتوي كل سجل على قيمة محددة للمفتاح الأساسي.
3. **يتم تعيينه تلقائيًا كمؤشر فهرسة (Index) في معظم أنظمة قواعد البيانات** مما يساهم في تحسين أداء البحث والاستعلامات.

مثال على المفتاح الأساسي:

لنفترض أن لدينا جدولًا باسم **الطلاب (Students)** يحتوي على الأعمدة التالية:

Student ID (PK)	Name	Age	Major
1	أحمد	20	هندسة الحاسوب
2	فاطمة	22	علوم البيانات
3	خالد	21	الذكاء الاصطناعي

في هذا الجدول، العمود **Student ID** هو المفتاح الأساسي لأنه يحدد كل طالب بشكل فريد.

ثانيًا: المفتاح الأجنبي(Foreign Key)

تعريف المفتاح الأجنبي:

المفتاح الأجنبي هو عمود أو مجموعة أعمدة في جدول تشير إلى المفتاح الأساسي في جدول آخر.
يُستخدم المفتاح الأجنبي لإنشاء العلاقات بين الجداول وضمان تكامل البيانات بينهما.

خصائص المفتاح الأجنبي:

1. **يشير إلى المفتاح الأساسي في جدول آخر** :مما يربط الجداول ببعضها.
2. **يمكن أن يحتوي على قيم مكررة** :لأن أكثر من سجل في الجدول الحالي قد يشير إلى نفس المفتاح الأساسي في الجدول المرتبط.
3. **يمكن أن يقبل القيم الفارغة (NULL) في بعض الحالات** :إذا لم تكن العلاقة إلزامية.
4. **يضمن تكامل البيانات** :بحيث لا يمكن إدراج قيمة غير موجودة في الجدول المرتبط.

مثال على المفتاح الأجنبي:

لنفترض أن لدينا جدولًا باسم **المقررات الدراسية: (Courses)**

Course ID (PK)	Course Name	Student ID (FK)
101	قواعد البيانات	1
102	الذكاء الاصطناعي	2
103	التعلم العميق	3
104	الشبكات	1

في هذا الجدول:

- العمود **Course ID** هو المفتاح الأساسي لأنه يحدد كل مقرر بشكل فريد.
- العمود **Student ID** هو مفتاح أجنبي يشير إلى المفتاح الأساسي في جدول الطلاب **(Student ID) في جدول (Students.**

المحاضرة 8 : طريقة تمثيل الانواع المختلفة للكيانات والسمات

يعد تمثيل الكيانات والسمات في قواعد البيانات أحد المفاهيم الأساسية في تصميم قواعد البيانات العلائقية. من خلاله، يتم تنظيم البيانات بطريقة فعالة لتمكين استرجاعها وإدارتها بسهولة.

أولًا: الكيانات(Entities)

تعريف الكيان:

الكيان هو أي شيء يمكن تمثيله في قاعدة البيانات ويكون له خصائص مميزة، مثل شخص، مكان، كائن، أو حدث. يُستخدم الكيان لتخزين المعلومات ذات الصلة به في قاعدة البيانات.

أنواع الكيانات:

1. **الكيانات القوية (Strong Entities):** هي الكيانات التي تمتلك مفتاحًا أساسيًا يميز كل سجل داخل الجدول.
2. **الكيانات الضعيفة (Weak Entities):** هي الكيانات التي لا يمكن تمييز سجلاتها دون الاعتماد على كيان آخر.

مثال على الكيان القوي:

لنفترض أن لدينا جدولًا باسم الطلاب: **(Students)**

Student ID (PK)	Name	Age	Major
1	أحمد	20	هندسة الحاسوب
2	فاطمة	22	علوم البيانات
3	خالد	21	الذكاء الاصطناعي

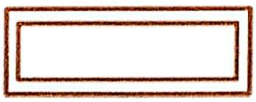
ويتم تمثيل الكيان ب

مثال على الكيان الضعيف:

لنفترض أن لدينا جدولًا باسم **العناوين (Addresses)** الذي يعتمد على جدول الطلاب:

Address ID	Street	City	Student ID (FK)
1	الشارع 1	بغداد	1
2	الشارع 2	البصرة	2
3	الشارع 3	الموصل	3

في هذا الجدول، الكيان **Addresses** يعتمد على **Students**، مما يجعله كيانًا ضعيفًا.

ويتم تمثيل الكيان الضعيف ب 

ثانيًا: السمات(Attributes)

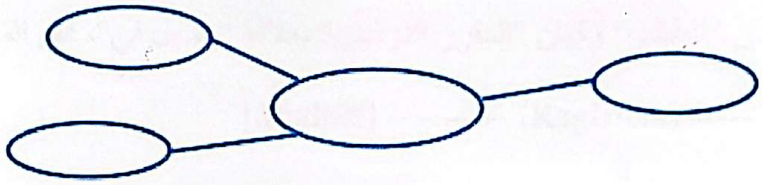
تعريف السمة:

السمة هي خاصية أو وصف يُستخدم لتمثيل معلومات حول كيان معين.

أنواع السمات:

1. **السمات البسيطة (Simple Attributes):** هي السمات التي تحتوي على قيمة واحدة لكل سجل، مثل "الاسم" و "العمر"

2. **السمات المركبة (Composite Attributes):** هي السمات التي يمكن تقسيمها إلى أجزاء أصغر، مثل "العنوان" الذي يتكون من "الشارع" و"المدينة."



3. **السمات المشتقة (Derived Attributes):** هي السمات التي يمكن حسابها بناءً على سمات أخرى، مثل "العمر" المحسوب من "تاريخ الميلاد

4. **السمات متعددة القيم (Multivalued Attributes):** هي السمات التي يمكن أن تحتوي على أكثر من قيمة لكل كيان، مثل "أرقام الهواتف."

مثال على السمات:

في جدول الطلاب، يمكن أن تكون السمات كالتالي:

- **سمة بسيطة:** الاسم(Name)
- **سمة مركبة:** العنوان(Address) ، الذي يتكون من الشارع، المدينة، الرمز البريدي
- **سمة مشتقة:** العمر(Age) ، المحسوب من تاريخ الميلاد(Date of Birth)
- **سمة متعددة القيم:** أرقام الهواتف (Phone Numbers)

ثالثًا: علاقة الكيانات والسمات في قاعدة البيانات

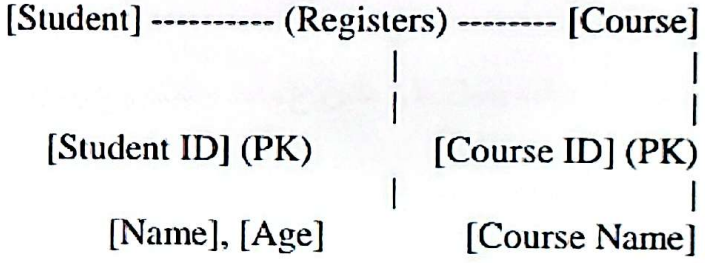
تمثيل الكيانات والسمات باستخدام مخطط ERD

يتم استخدام مخطط العلاقة بين الكيانات (ERD) لتمثيل الكيانات، السمات، والعلاقات بينها. في هذا المخطط:

- يتم تمثيل الكيانات كمستطيلات.
- يتم تمثيل السمات كأشكال بيضاوية.
- يتم تمثيل المفاتيح الأساسية بتسطير السمة الرئيسية.
- يتم توصيل الكيانات بالعلاقات عبر خطوط.

مثال على مخطط:ERD

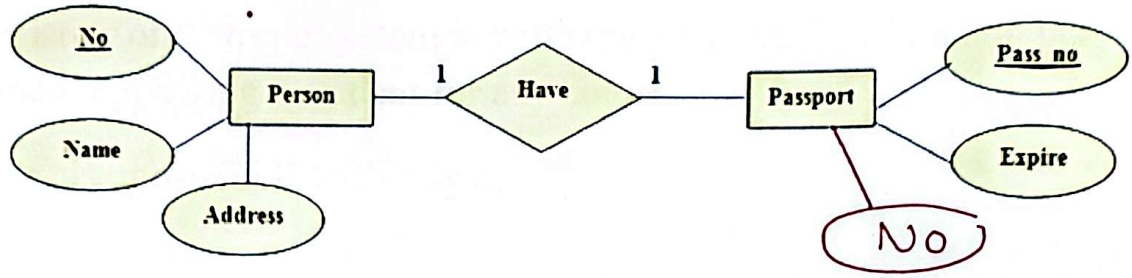
إذا كان لدينا كيان "الطالب" وكيان "المقرر الدراسي" بعلاقة "يسجل في"، فإن المخطط سيكون كالتالي:



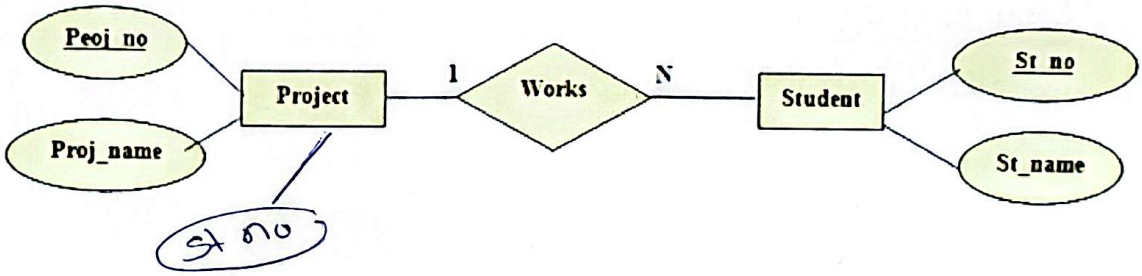
لمحاضرة 9: طريقة رسم ERD

مثال: علاقة الاشخاص مع جوازات السفر

كل شخص لديه جواز سفر واحد وكل جواز سفر يعود لشخص واحد فقط.



مثال: كل طالب يعمل على مشروع واحد والمشروع يعمل عليه اكثر من طالب.



مثال: الطلاب مع المقررات الدراسية.....

امثلة عن الاسئلة بالامتحانات:

السؤال الاول:

Suppose you have a (Learning management System), this system have a number of students those registered to many courses, The registrar of learning system is write the year and the class number when registering a student for any course.

- Propose ERD to this system.

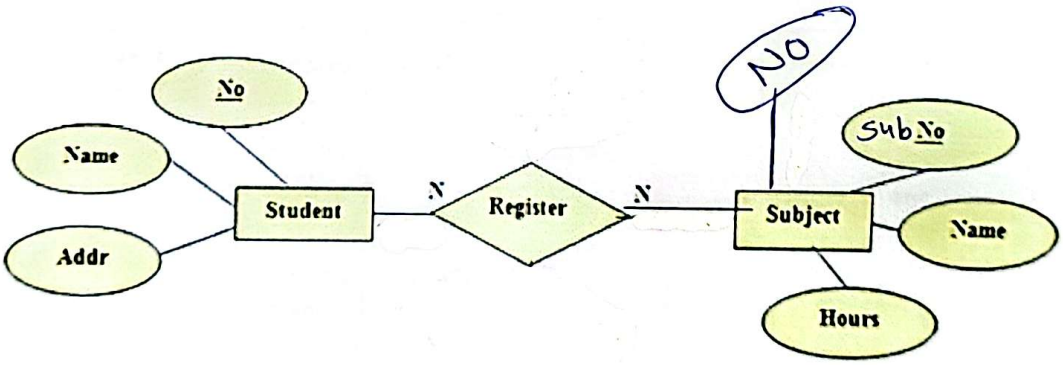
تحديد الكيانات:

الطالب (اسم الطالب – الرقم الجامعي – العنوان)

المقرر (اسم المقرر – رقم المقرر – عدد الساعات – NO)

تحديد العلاقات:

علاقة تسجيل الطالب لمقرر.



السؤال الثاني:

Suppose we have a (Company management System), this company contains many Employees that are belongs to Departments, and works on Projects. One department have many employees, and every employee is works to one project, as well as every one project is worked by many employees.

- Propose ER diagram to this system.

الحل:

الكيانات:

الموظف (رمز الموظف، اسم الموظف)

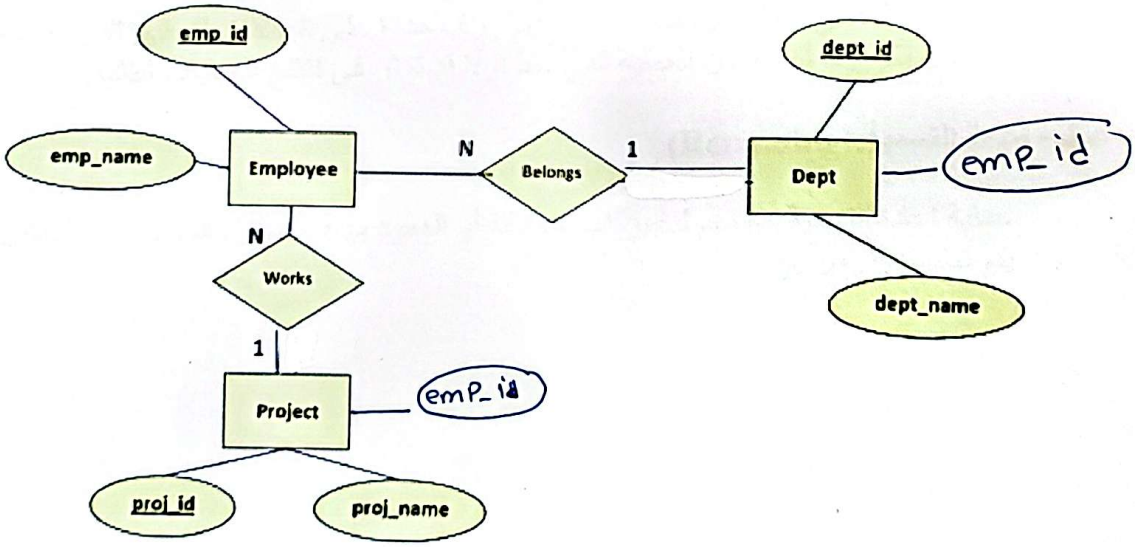
القسم (رمز القسم، اسم القسم، empid)

المشروع (رمز المشروع، اسم المشروع)

العلاقات:

علاقة الموظف مع القسم

علاقة الموظف مع المشروع



المحاضرة 10: الجبر العلاقي - الجزء الأول (Relational Algebra - Part 1)

العمليات الأحادية SELECT وPROJECT:

عملية:SELECT

- **تعريف العملية:**
  - عملية **SELECT** تُستخدم لاستخراج بيانات من قاعدة البيانات تُطابق شروط معينة. يُشار إليها بالحرف الكبير."σ"
  - هي عملية أحادية تُطبق على علاقة واحدة فقط.

عملية:PROJECT

- **تعريف العملية:**
  - عملية **PROJECT** تُستخدم لاختيار أعمدة محددة من العلاقة.
  - هي عملية أحادية تُطبق على علاقة واحدة فقط.

تسلسل العمليات وعملية إعادة التسمية

تسلسل العمليات:

- في الجبر العلاقي، يُمكن ترتيب العمليات بصورة محددة على العلاقات لتحقيق النتيجة المطلوبة.
- التسلسل يكون ذا أهمية لأن العملية التي تُنفذ أولًا قد تؤثر في نتائج العمليات التالية.

عملية إعادة التسمية:(Renaming)

- عملية إعادة التسمية تُستخدم لتغيير اسم العلاقة أو العمود بهدف تسهيل المعالجة أو التحليل.
- يتم تمثيلها بالرمز .ρ

المحاضرة 11: الجبر العلائقي - الجزء الثاني والثالث

العمليات الثنائية

1. عملية:UNION

- **تعريف العملية:**
  - عملية UNIONتُستخدم لدمج مجموعتين من البيانات من نفس النوع.
  - تُعيد كل السجلات من المجموعتين دون تكرار.
  - العملية تكون متوافقة بين المجموعات (يجب أن تحتوي المجموعات على نفس الأعمدة بنفس الترتيب والنوع).
- **التركيب الأساسي:**
  - **R U S:** تُعني دمج المجموعتين R و S.

2. عملية:INTERSECTION

- **تعريف العملية:**
  - عملية **INTERSECTION** تُستخدم لاستخراج السجلات المشتركة بين مجموعتين.
- **التركيب الأساسي:**
  - **R ∩ S:** تُعني استخراج السجلات المشتركة بين R و S.

3. عملية:MINUS

- **تعريف العملية:**
  - عملية **MINUS** تُستخدم لاستخراج السجلات الموجودة في مجموعة واحدة ولا توجد في الأخرى.
- **التركيب الأساسي:**
  - **R - S:** تُعني استخراج السجلات التي في R ولا توجد في S.

علاقات الانضمام(JOIN)

1. عملية:INNER JOIN

- **تعريف العملية:**
  - **INNER JOIN** تُستخدم لدمج البيانات من جداول مختلفة بناءً على علاقة مشتركة بين الأعمدة.
  - تُعيد فقط السجلات التي لها تطابق بين الجداول.
- **التركيب الأساسي:**
  - **R ⋈ S:** تُعني دمج الجداول R و S بناءً على علاقة مشتركة بينهما.

2. عملية:OUTER JOIN
   - تعريف العملية:
     - OUTER JOIN تُستخدم لدمج البيانات بين الجداول مع إبقاء السجلات التي لا تحتوي على تطابق بين الجداول.
     - يوجد ثلاثة أنواع من :OUTER JOIN
       - LEFT OUTER JOIN: تُعيد جميع السجلات من الجدول الأيسر و السجلات المتوافقة من الجدول الأيمن.
       - RIGHT OUTER JOIN: تُعيد جميع السجلات من الجدول الأيمن و السجلات المتوافقة من الجدول الأيسر.
       - FULL OUTER JOIN: تُعيد جميع السجلات من كلا الجدولين، مع تضمين السجلات غير المتوافقة من كلا الجانبين.
   - التركيب الأساسي:
     - R ⟕ S (LEFT OUTER JOIN)
     - R ⟖ S (RIGHT OUTER JOIN)
     - R ⟗ S (FULL OUTER JOIN)

3. عملية القسمة(DIVISION)

1. تعريف العملية:
   - DIVISION تُستخدم لاستخراج السجلات التي تحتوي على قيم موجودة في جميع السجلات المرتبطة بجدول آخر.
   - غالبًا ما يتم استخدامها عندما نريد العثور على البيانات التي تتوافق مع جميع السجلات في مجموعة أخرى.
   - التركيب الأساسي:
     - R ÷ S: تُعني استخراج السجلات التي تظهر في جميع السجلات في مجموعة Sمن مجموعة R.

المحاضرة 12-15: الملفات والسجلات (Files and Records)

Records and Record Types

السجلات:(Records)

- السجل هو وحدة تخزين بيانات ضمن ملف، يمثل كيانًا معينًا مثل موظف أو طالب.
- كل سجل يحتوي على مجموعة من الحقول (Fields) التي تمثل خصائص الكيان.

أنواع السجلات:(Record Types)

- **سجلات ذات طول ثابت:(Fixed Length Records)**
  - كل سجل له نفس عدد الحقول بنفس الطول.
  - **مثال :**سجل يحتوي على الحقول المعرف :10 أحرف، اسم: 20 حرفًا، عمر: 3 أرقام.
- **سجلات ذات طول متغير:(Variable Length Records)**
  - كل سجل قد يحتوي على عدد مختلف من الحقول أو أطوال مختلفة للحقل الواحد.
  - **مثال :**سجل يحتوي على الحقول :المعرف 10 :أحرف، اسم: نص متغير الطول.

Formatting Records تنسيق السجلات

تنسيق سجلات ذات طول ثابت:

- كل حقل يأخذ مساحة محددة مسبقًا.
- مثال: سجل الموظف(ID: 10 ، الاسم: 20، العمر: 3) سيشغل 33 بايت.

تنسيق سجلات ذات طول متغير:

- يتم استخدام تقنيات خاصة لتحديد نهاية كل حقل أو نهاية السجل:
  - **فاصل الحقول :(Field Separator)** مثل استخدام رمز (|) لفصل الحقول.
  - **الفواصل الثنائية :(Binary Indicators)** لإظهار بداية ونهاية الحقول.
  - **أطوال الحقول :(Field Lengths)** يتم تخزين طول الحقل قبل قيمته.

تنسيق ملفات تحتوي على سجلات اختيارية:

- يمكن أن تحتوي السجلات على حقول قد تكون موجودة أو غير موجودة. يتم استخدام علامة (مثل ("Null" لتحديد الحقول غير الموجودة.

تنسيق الحقول المتكررة:

- تُستخدم لتخزين بيانات مثل أرقام الهواتف المتعددة لشخص واحد. يتم فصل القيم المتكررة باستخدام علامات مميزة.

تنسيق ملفات تحتوي على أنواع مختلفة من السجلات:

- يتم استخدام معرف السجل (Record Identifier) لتحديد نوع السجل.
  - **مثال :**سجل موظف(Type=1) ، سجل عميل.(Type=2)

Organizing Records in the File تنظيم السجلات داخل الملف

Record Blocking تقسيم السجلات إلى كتل

- **السجلات المجمعة:(Blocking)**
  - تُجمع السجلات في كتل لتقليل المساحة المستخدمة وتسريع الوصول.
  - الكتلة قد تحتوي على سجل واحد أو أكثر.

Spanned vs Unspanned Records السجلات الممتدة وغير الممتدة

- **السجلات غير الممتدة:(Unspanned Records)**
  - يتم تخزين كل سجل في كتلة واحدة فقط. إذا كان السجل كبيرًا جدًا، يتم رفضه.
- **السجلات الممتدة:(Spanned Records)**
  - إذا كان السجل أكبر من الكتلة، يتم تقسيمه عبر كتل متعددة.

Organizing Files on Disk تنظيم الملفات على القرص

Allocating File Blocks on Disk تخصيص كتل الملفات على القرص

- **التخصيص المتتابع:(Contiguous Allocation)**
  - يتم تخزين جميع كتل الملف بشكل متتابع على القرص.
  - **المميزات :**وصول سريع.
  - **العيوب :**صعوبة التوسع.
- **التخصيص المرتبط:(Linked Allocation)**
  - يتم تخزين كل كتلة مع مؤشر يشير إلى الكتلة التالية.
  - **المميزات :**مرونة.
  - **العيوب :**وصول أبطأ.
- **التخصيص المفهرس:(Indexed Allocation)**
  - يتم استخدام جدول يحتوي على مواقع الكتل.
  - **المميزات :**سرعة الوصول.
  - **العيوب :**يحتاج مساحة إضافية للفهرسة.

Files and File Headers الملفات ورؤوس الملفات

File Headers رؤوس الملفات

- تحتوي على بيانات وصفية عن الملف مثل:
  - عدد السجلات.
  - طول السجل.
  - طريقة التخصيص.

Files of Unordered Records (Heap Files)

- السجلات يتم تخزينها دون ترتيب.
- **المميزات :**سرعة الإدراج.
- **العيوب :**البحث قد يكون بطيئًا.

Files of Ordered Records (Sorted Files)

- السجلات يتم تخزينها بترتيب معين (مثل الترتيب الأبجدي.(
- **المميزات :**سرعة البحث الثنائي.
- **العيوب :**إدراج السجلات أبطأ.

Hashing Techniques تقنيات التجزئة

1. Hash Table جداول التجزئة

- تُستخدم لتسريع عمليات البحث باستخدام مفتاح.
- يتم حساب موقع السجل باستخدام دالة تجزئة.(Hash Function)

The Idea Behind Hashing فكرة التجزئة

- يتم تطبيق دالة رياضية على المفتاح لتحديد موقع السجل.

Hashing Functions دوال التجزئة

- **التجزئة المباشرة:(Direct Hashing)**
  - المفتاح يُستخدم مباشرة كموقع.
- **التجزئة باستخدام الطرح:(Subtraction Hashing)**
  - يتم طرح قيمة ثابتة من المفتاح.
- **التجزئة باستخدام القسمة:(Modulo Division Hashing)**
  - يتم قسمة المفتاح على حجم الجدول، والموقع هو الباقي.

Good Luck