



المملكة العربية السعودية
المؤسسة العامة للتدريب المهني والتقني
الادارة العامة للتدريب المهني والأهلي
معهد التميز الكندي العالي للتدريب

برمجة قواعد البيانات



لغة SQL المتقدمة

مقدمة SQL

مقدمة

- تعريف لغة الاستعلام المهيكلة SQL

- الهدف من دراسة مادة SQL

-- مفهوم قواعد البيانات :

- تُعرّف قاعدة البيانات بأنها عبارة عن كمية كبيرة من البيانات والمعلومات التي يُمكن الرجوع إليها وإجراء العمليات المختلفة عليها كعمليات البحث والتعديل والمقارنة، وذلك من خلال تكلفة بسيطة وسرعة عالية، وتمتاز هذه البيانات الرقمية بأنها بيانات ذات فهرسة وترتيب معين، وتُعتبر لغة SQL هي اللغة الأكثر استخداماً وشيوعاً في برمجة قواعد البيانات وتتكوّن قاعدة البيانات من جدول واحد أو أكثر، مثل سجل الخاص بالموظف الذي يتكوّن من عدّة حقول، مثل: رقم الموظف، واسم الجهاز، ودرجة الموظف، وتاريخ التعيين، والراتب، وبيانات الموظف التي تخزن في جهاز الحاسوب تكون على نحو منظم، حيث يسهل لنا الحاسوب التعامل مع البيانات والبحث ضمن هذه البيانات، والتمكين من الإضافة والتعديل.

مميزات قواعد البيانات

1/ سرعة الوصول والاسترجاع للبيانات

2/ توفير المساحات الهائلة التي تحتلها مكاتب الأرشيف.

3/ تكامل البيانات

4/ السهولة في مشاركة البيانات الرقمية مقارنة بالبيانات الورقية.

5/ السرية والأمان في حفظ البيانات

6/ تجميع كافة البيانات بطريقة مفهومة

7/ إمكانية إجراء أي تعديلات على هذه القواعد، ومراقبة التغييرات التي قد تحدث عليها.

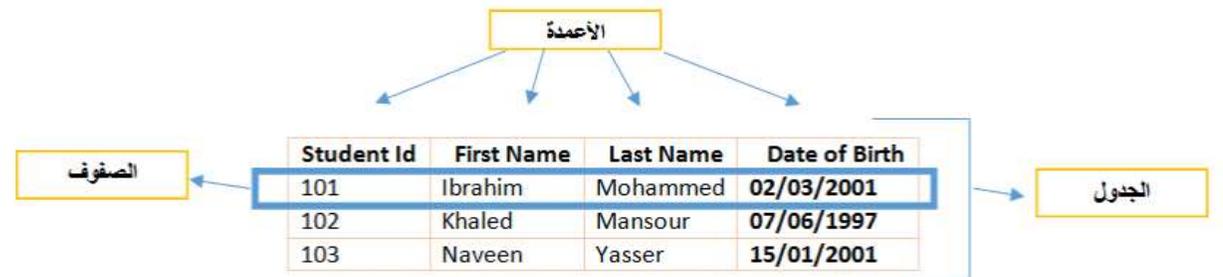
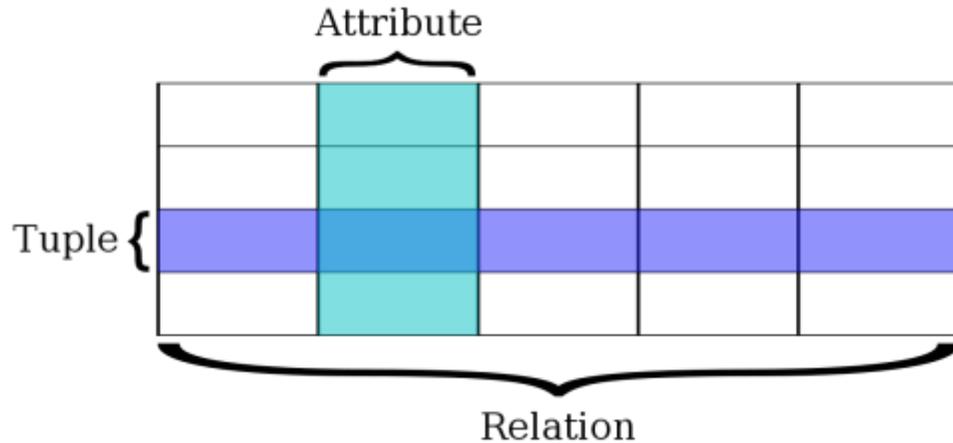
قواعد البيانات العلائقية

تتكون من:

1/ مجموعة المعطيات المستخدمة في قاعدة البيانات

2/ توابع تساعد على الوصول الى البيانات و تعديلها و استرجاعها

3/ مجموعة من القواعد التي تنظم العمليات التي يمكن اجرائها على قاعدة البيانات



أنظمة إدارة قواعد البيانات العلائقية DBMS

هو عبارة عن مجموعة برامج حاسوبية تتحكم في تنظيم وتخزين وإدارة وسحب البيانات (المعطيات) من قاعدة بيانات. يدير النظام العديد من قواعد البيانات كما يمكن العديد من المستخدمين من الوصول إلى هذه القواعد في الوقت نفسه

- مكونات نظام إدارة قواعد البيانات

1/ محرك قاعدة البيانات

2/ دليل قاعدة البيانات

3/ واجهات مرئية

4/ أدوات خاصة

5/ أدوات تطوير التطبيقات

تختلف قواعد البيانات عن أنظمة إدارة قواعد البيانات

SQL

تقوم على مبدا توجيه طلب الى محرك قاعدة البيانات و الحصول على جواب او مجموعة اجوبة

اللغات الفرعية لتعليمات الـ SQL :

لغة معالجة البيانات : `select, insert, delete, update`

لغة تعريف البيانات : `create table, drop table, alter table, create index`

لغة التحكم بالبيانات : `grant, revoke`

لغة معالجة البيانات

الدرس الثاني

الكلمات المفتاحية

التعليمة	استخدامها
SELECT	لاستعادة البيانات من قاعدة البيانات
INSERT	لإدراج سجلات ضمن جدول في قاعدة البيانات
DELETE	لحذف سجل أو مجموعة من السجلات
UPDATE	لتعديل سجل أو مجموعة من السجلات

تعليمية SELECT

تأخذ الصيغة العامة :

```
SELECT field1, field2,.. FROM table-name;
```

تستخدم لاسترجاع حقل محدد من جدول محدد اما اذا اردنا استرجاع كافة الحقول نستخدم الرمز * كبديل لاسم الحقول

مثال: `Select * from student`

Serial_Num	Fname	Lname	Stud_ID	Birth_Year	City	Gender
1	Sarah	Mohamed	48751	1994	Najran	Female
2	Mona	Ali	92001	1993	Abha	Female
3	Huda	Ahmad	71911	1995	Najran	Female
4	Ali	Saleh	69159	1991	Jeddah	Male
5	Sarah	Saud	84331	1994	Abha	Female
6	Abdulaziz	Ahmad	11043	1994	Najran	Male

SELECT

- تعبير **distinct** يستخدم لاستعادة جميع السجلات مع الغاء التكرار

مثال: `Select DISTINCT Fname FROM student`

Fname

Sarah

Mona

Huda

Ali

Abdulaziz

SELECT

- تعبير **ORDER BY** لترتيب السجلات حسب ترتيب تصاعدي (ASC) او تنازلي (DESC)

```
Select Serial_num FROM `student` ORDER BY Serial_num DESC
```

Serial_num
6
5
4
3
2
1

SELECT

- تعبير AS في حال الرغبة باستخدام اسماء بديلة لحقول جدول القيم المعادة

Select Fname AS Names FROM student

Names

Sarah

Mona

Huda

Ali

Sarah

Abdulaziz

SELECT

- تعبير WHERE تأخذ الصيغة العامة :

```
SELECT * FROM table_name WHERE condition
```

- تعبير LIKE يستخدم ضمن العبارة الشرطية كشرط لوجود مثل و تستخدم عادة مع علامة %

```
SELECT * FROM table_name WHERE field1 LIKE '%A%'
```

A تحتوي على حرف

```
SELECT * FROM table_name WHERE field1 LIKE 'A'
```

A تنتهي بحرف

```
SELECT * FROM table_name WHERE field1 LIKE 'A%'
```

A تبدأ بحرف

SELECT

```
SELECT * FROM `student` WHERE Fname LIKE 'A%'
```

Serial_Num	Fname	Lname	Stud_ID	Birth_Year	City	Gender
4	Ali	Saleh	69159	1991	Jeddah	Male
6	Abdulaziz	Ahmad	11043	1994	Najran	Male

SELECT

تعبير BETWEEN ضمن العبارة الشرطية كشرط لوجود قيمة محصورة بين قيمتين محددتين

```
SELECT * FROM table_name WHERE field1 BETWEEN .. (AND/OR)...
```

حيث ان AND ,OR هي عمليات منطقية (و , أو) تستخدم مع الشرط ويمكن استخدام كلمة NOT للنفي

EXAMPLE:

```
SELECT * FROM `student` WHERE Serial_Num BETWEEN 2 AND 5
```

Serial_Num	Fname	Lname	Stud_ID	Birth_Year	City	Gender
2	Mona	Ali	92001	1993	Abha	Female
3	Huda	Ahmad	71911	1995	Najran	Female
4	Ali	Saleh	69159	1991	Jeddah	Male
5	Sarah	Saud	84331	1994	Abha	Female

HOME WORK

أوجدني ناتج الاوامر التالية:

1/ SELECT * FROM `student` WHERE Fname LIKE '%A'

2/ SELECT * FROM `student` WHERE Fname LIKE '%A%'

3/ SELECT * FROM `student` WHERE Fname LIKE '%A%'

AND Serial_Num BETWEEN 2 AND 5

HOME WORK

أكتبني امر للأستعلام عن الحقول التالية (Serial_Num, Fname, Stud_ID) من جدول اسمه student

أكتبني امر أستعلام حسب الناتج المعطى في الصورة التالية:

Lname	Stud_ID
Mohamed	48751
Ali	92001
Ahmad	71911
Saleh	69159
Saud	84331
Ahmad	11043

SQL

تابع لغة معالجة البيانات+التوابع
الدرس 3

Delete

تعليمة delete تقوم بحذف سجل او مجموعة سجلات وتأخذ الصيغة العامة:

Delete from table_name;

Delete from student

تقوم بحذف جميع السجلات من الجدول

Delete from student where Fname='sarah'

insert

تعليمة insert تستخدم لادراج سجل في جدول محدد و تاخذ الضيغة العامة:

- 1/
Insert into table_name values (value1, value2, ..);
- 2/
Insert into table_name (field1, field2...)
values (value1, value2, ..);

insert

Insert into student values (7 , 'Ahmad','fahad',55301,1996,'Abha','Male')

Serial_Num	Fname	Lname	Stud_ID	Birth_Year	City	Gender
1	Sarah	Mohamed	48751	1994	Najran	Female
2	Mona	Ali	92001	1993	Abha	Female
3	Huda	Ahmad	71911	1995	Najran	Female
4	Ali	Saleh	69159	1991	Jeddah	Male
5	Sarah	Saud	84331	1994	Abha	Female
6	Abdulaziz	Ahmad	11043	1994	Najran	Male
7	Ahmad	fahad	55301	1996	Abha	Male

update

تستخدم في تعديل البيانات في سجل او في مجموعة من السجلات

Update table_name Set

Field1 = new_field_value1,

Field2= new_field_value2,

Where condition;

update

UPDATE student **SET** City='Najran' **WHERE**
Serial_Num=7

Serial_Num	Fname	Lname	Stud_ID	Birth_Year	City	Gender
1	Sarah	Mohamed	48751	1994	Najran	Female
2	Mona	Ali	92001	1993	Abha	Female
3	Huda	Ahmad	71911	1995	Najran	Female
4	Ali	Saleh	69159	1991	Jeddah	Male
5	Sarah	Saud	84331	1994	Abha	Female
6	Abdulaziz	Ahmad	11043	1994	Najran	Male
7	Ahmad	fahad	55301	1996	Najran	Male

التوابع

هو عبارة عن تعبير رياضي يأخذ مجموعة من القيم المدخلة المدعوة بالمعاملات

أنواع التوابع في الـ SQL

- التوابع التجميعية : هي توابع تأخذ كمعاملات مجموعة من القيم وتعيد قيمة وحيدة مثل التابع الذي يحسب مجموع اعداد حقيقية
- التوابع الدرجية : وهي التوابع التي تأخذ معاملا وحيدا وتعيد قيمة وحيدة مثل تابع القيمة المطلقة

التوايع التجميعية

التابع	استخدامه
AVG	يقوم بحساب معدل القيم لحقل معين
COUNT	يقوم بحساب عدد البيانات الخاصة بحقل معين
MIN	يقوم باعادة القيمة الصغرى من قيم حقل معين
MAX	يقوم باعادة القيمة العظمى من قيم حقل معين
SUM	يقوم بحساب مجموع قيم حقل معين

التابع AVG

يستخدم لحساب معدل القيم لحقل معين

الصيغة : `Select AVG (column_name) from table_name`

مثال : `SELECT AVG (Marks) FROM grades`

student_ID	Name	Subject	Marks
1	Sara	Math	89
2	Huda	Math	85
3	Lamia	Math	93
4	Suad	Math	96

AVG (Marks)

90.7500

التابع COUNT

يستخدم في حساب عدد البيانات في الجدول من اجل حقل معين

```
SELECT COUNT(Marks) FROM `grades`
```

student_ID	Name	Subject	Marks
1	Sara	Math	89
2	Huda	Math	85
3	Lamia	Math	93
4	Suad	Math	96

COUNT(Marks)

4

التابع MIN & MAX

يستخدم لإعادة القيمة الصغرى والعظمى

```
SELECT MIN(Marks) FROM `grades`
```

```
SELECT MAX(Marks) FROM `grades`
```

student_ID	Name	Subject	Marks
1	Sara	Math	89
2	Huda	Math	85
3	Lamia	Math	93
4	Suad	Math	96

MIN(Marks)

85

MAX(Marks)

96

التابع SUM

تستخدم في حساب مجموع قيم حقل معين

```
SELECT SUM(Marks) FROM `grades`
```

student_ID	Name	Subject	Marks
1	Sara	Math	89
2	Huda	Math	85
3	Lamia	Math	93
4	Suad	Math	96

SUM(Marks)

363

قواعد بيانات

الدرس 4

تجميع البيانات

تعليلة GROUP BY تستخدم لتجميع البيانات وتاخذ الصيغة العامة:

```
SELECT `field1`, `field2` FROM `table_name` GROUP BY `field1`;
```

EXAMPLE:

```
SELECT `gender` FROM `student` GROUP BY `gender`
```

Gender

Female

Male

```
SELECT `city` FROM `Student` GROUP BY `city`
```

تجميع البيانات

```
SELECT `city` FROM `Student` GROUP BY `city`
```

city

Abha

Jeddah

Najran

```
SELECT `name`,SUM(`marks`) FROM `grades` GROUP BY `name`
```

name	SUM('marks')
Huda	85
Lamia	93
Sara	89
Suad	96

HAVING كلمة

يستخدم عادة مع الكلمة المفتاحية `group by` لتصفية المجموعة حسب شرط محدد أي انها مخصصة لوضع شروط وتأخذ الصيغة العامة:

```
SELECT field1,..  
FROM table_name  
WHERE condition  
GROUP BY column_name(s)  
HAVING condition  
ORDER BY column_name(s);
```

HAVING كلمة

Example:

```
SELECT `subject`,AVG(`marks`) FROM `grades` GROUP BY `subject` HAVING AVG(`marks`) >60
```

subject	AVG(`marks`)
Math	90.7500

```
SELECT `subject`,AVG(`marks`) FROM `grades` GROUP BY `subject` HAVING AVG(`marks`) >95
```

subject	AVG(`marks`)
---------	--------------

كلمة top N

يستخدم لاسترجاع اول N من سجلات الاستعلام و تاخذ الصيغة العامة:

```
SELECT field1, field2,...  
FROM table_name  
WHERE condition  
LIMIT 0,N ;
```

ملاحظة: 0 هو سجل البداية و N هو سجل النهاية و كلمة limit هي مجال الارقام

top N كلمة

Example:

```
SELECT `student_ID` FROM `grades` LIMIT 0,2
```

student_ID

1

2

التوابع الدرجية

هي التوابع التي تاخذ معاملا وحيدا و تعيد قيمة وحيدة و تصنف الى 4 اصناف اساسية

تعريفه	نوع التابع
توابع خاصة بالعمليات على الارقام	التوابع الرقمية
توابع خاصة بالعمليات على سلاسل المحارف	توابع سلاسل المحارف
توابع خاصة بالعمليات على التاريخ و الوقت	توابع التاريخ و الوقت
توابع خاصة بعملية تحويل معامل الدخل من نمط بيانات الى اخر	توابع التحويل

التوابع الرقمية

هي التوابع الدرجية الخاصة بالعمليات على القيم الرقمية

التابع	استخدامه	مثال
Floor()	قرب مُعامل الدخل إلى أقرب عدد صحيح أصغر من مُعامل الدخل	<code>select floor(3.6)=3</code>
Ceiling()	قرب مُعامل الدخل إلى أقرب عدد صحيح أكبر من مُعامل الدخل	<code>select ceiling(3.6)=4</code>
Round()	يُقرب مُعامل الدخل ذو الفاصلة العشرية إلى أقرب عدد صحيح بدقة محددة	<code>select round(3.6)=4</code>
Abs()	عيد القيمة المطلقة لمُعامل الدخل	<code>select abs(-3) =3</code>
Sin(), cos(), tan(), atan()	توابع تحسب قيم ظل، تظل، جب، تجب الزاوية	<code>select cos(70) =0.6333</code>
Sqrt()	عيد قيمة الجذر التربيعي لمُعامل الدخل	<code>select sqrt(49) =7</code>
Rand()	يُعيد رقم عشوائي بين 0 و 1 و يستخدم لتوليد الرقم العشوائي	<code>select rand() 0.8364907566797=</code>

قواعد بيانات

الدرس 5

توابع سلاسل الحروف

هي التوابع الخاصة بالعمليات على سلاسل المحارف

التابع	استخدامه	مثال
LEFT()	يُعيد جزء من السلسلة يبتدئ من بدايتها حتى عدد محدد من المحارف	<pre>SELECT LEFT('hello world',4)</pre> hell
RIGHT()	يُعيد جزء من السلسلة يبتدئ من نهايتها حتى عدد محدد من المحارف	<pre>SELECT RIGHT('hello world',4)</pre> orld
SUBSTR()	يُعيد جزء من السلسلة يبتدئ من موقع محدد فيها وبطول عدد محدد من المحارف	<pre>SELECT substr('hello world',1,7)</pre> Hello w
LENGTH()	يُعيد طول السلسلة المحرفية	<pre>SELECT length('hello world')</pre> 11

توابع سلاسل الحروف

هي التوابع الخاصة بالعمليات على سلاسل المحارف

التابع	استخدامه	مثال
CONCAT()	يُستخدم لدمج أكثر من سلسلة محرفية	<pre>SELECT concat('hello',' ','world') Hello world</pre>
LOWER() UPPER()	يحوّل جميع محارف السلسلة إلى أحرف كبيرة أو صغيرة	<pre>SELECT upper('hello') =HELLO SELECT lower('HELLO')=hello</pre>
TRIM()	يلغي الفراغات من بداية ونهاية السلسلة المحرفية	<pre>SELECT TRIM(' hello ') 'hello'</pre>
INSTR()	يُستخدم لتحديد موقع سلسلة جزئية ضمن سلسلة رئيسية	<pre>SELECT instr('hello','lo') 4</pre>

توابع التاريخ والوقت

هي التوابع الخاصة بالعمليات على التاريخ والوقت

التابع	استخدامه	مثال
DATEDIFF()	يُعيد الفرق بين تاريخين	
GETDATE()	يُعيد السنة، والتاريخ، واليوم،	
CURRENT_DATE()	يُعيد التاريخ الحالي الخاص بنظام إدارة قاعدة البيانات والساعة، والدقيقة، والثانية، وأجزاء الثانية	
CURRENT_TIME()	يُعيد التوقيت الخاص بنظام إدارة قاعدة البيانات	
CURRENT_TIMESTAMP()	يُعيد التاريخ والتوقيت الخاصين بنظام إدارة قاعدة البيانات	

توابع التحويل

هي التوابع الدرجية الخاصة بالتحويل من نمط بيانات الى نمط اخر

التابع	استخدامه	مثال
Str()	يحول قيمة الدخل العددية إلى سلسلة حرفية	SELECT STR(185.5) = 186
To_Number()	يحول سلسلة المحارف المارة كمعامل دخل إلى عدد	
Cast()	يحول قيمة الدخل إلى قيمة من أي نمط آخر من البيانات	SELECT CAST(25.65 AS int) = 25
Convert()	يحول قيمة الدخل إلى قيمة من أي نمط آخر من البيانات	

قواعد بيانات

الدرس 6

الاستعلامات الفرعية

الاستعلام الفرعي هو أي استعلام يتم تضمينه في استعلام آخر ولكنها لا تدعم قواعد البيانات mysql بل تستعويض عنها بjoin وليس من الضرورة استخدام نفس الجداول في الاستعلامات الفرعية و الاستعلامات الرئيسية ولها نوعان وهي:

1/ استعلام فرعي مرتبط : تعتمد في عملها على استعلامات رئيسية حاوية لها

يتم فيها تكرار عملية تنفيذ الاستعلام بعدد مرات مساو لعدد السجلات التي يعيدها

2/ استعلام فرعي غير مرتبط (مستقل): تكون مستقلة عن الاستعلامات الرئيسية الحاوية لها

ينفذ الاستعلام ثم يمرر القيمة الى الاستعلام الرئيسي

الربط الداخلي والخارجي

الاستعلام عن أكثر من جدول في وقت واحد باستخدام الربط وللربط أنواع:

1 / inner join الربط الداخلي

2 / left join الربط اليساري

3 / right join الربط اليميني

4 / full join الربط التام

قاعدة البيانات

Student

ROLL_NO	NAME	ADDRESS	PHONE	Age
1	HARSH	DELHI	XXXXXXXXXX	18
2	PRATIK	BIHAR	XXXXXXXXXX	19
3	RIYANKA	SILIGURI	XXXXXXXXXX	20
4	DEEP	RAMNAGAR	XXXXXXXXXX	18
5	SAPTARHI	KOLKATA	XXXXXXXXXX	19
6	DHANRAJ	BARABAJAR	XXXXXXXXXX	20
7	ROHIT	BALURGHAT	XXXXXXXXXX	18
8	NIRAJ	ALIPUR	XXXXXXXXXX	19

StudentCourse

COURSE_ID	ROLL_NO
1	1
2	2
2	3
3	4
1	5
4	9
5	10
4	11

Join (Inner join)

الربط بالتساوي يستخدم في الربط بين اكثر من جدول ويكون الربط بين سجلات الجدول الاول و سجلات جدول ثاني اعتمادا على مساواة بين قيمة حقل او سجل بين الجدولين .

تاخذ الصيغة العامة :

```
SELECT table1.column1,table1.column2,table2.column1,....
```

```
FROM table1 INNER JOIN table2
```

```
ON table1.matching_column = table2.matching_column;
```

Join (Inner join)

```
SELECT StudentCourse.COURSE_ID, Student.NAME, Student.AGE
```

```
FROM Student INNER JOIN StudentCourse
```

```
ON Student.ROLL_NO = StudentCourse.ROLL_NO;
```

COURSE_ID	NAME	Age
1	HARSH	18
2	PRATIK	19
2	RIYANKA	20
3	DEEP	18
1	SAPTARHI	19

Left join (Left outer join)

تعيد جميع قيم الحقل التابع للجدول الاول والقيم المطابقة لها الخاصة بالحقل التابع للجدول الثاني ويتم ادراج null في قيم الحقل التابع للجدول الثاني في السجلات التي لا يوجد لها قيم مطابقة

عدد السجلات المعادة تكون مطابقة لعدد السجلات في الجدول الاول

تأخذ الصيغة العامة في الكتابة:

```
SELECT table1.column1,table1.column2,table2.column1,....
```

```
FROM table1 LEFT JOIN table2
```

```
ON table1.matching_column = table2.matching_column;
```

Left join (Left outer join)

```
SELECT Student.NAME, StudentCourse.COURSE_ID
```

```
FROM Student LEFT JOIN StudentCourse
```

```
ON StudentCourse.ROLL_NO = Student.ROLL_NO;
```

NAME	COURSE_ID
HARSH	1
PRATIK	2
RIYANKA	2
DEEP	3
SAPTARHI	1
DHANRAJ	NULL
ROHIT	NULL
NIRAJ	NULL

Right join (right outer join)

تعيد جميع قيم الحقل التابع للجدول الثاني والقيم المطابقة لها الخاصة بالحقل التابع للجدول الاول ويتم ادراج null في قيم الحقل التابع للجدول الاول في السجلات التي لا يوجد لها قيم مطابقة

عدد السجلات المعادة تكون مطابقة لعدد السجلات في الجدول الثاني

تأخذ الصيغة العامة في الكتابة:

```
SELECT table1.column1,table1.column2,table2.column1,....
```

```
FROM table1 RIGHT JOIN table2
```

```
ON table1.matching_column = table2.matching_column;
```

Right join (right outer join)

```
SELECT Student.NAME, StudentCourse.COURSE_ID
```

```
FROM Student RIGHT JOIN StudentCourse
```

```
ON StudentCourse.ROLL_NO = Student.ROLL_NO;
```

NAME	COURSE_ID
HARSH	1
PRATIK	2
RIYANKA	2
DEEP	3
SAPTARHI	1
NULL	4
NULL	5
NULL	4

Full join

تعيد جميع قيم الحقول التابع للجدول الاول والقيم المطابقة لها الخاصة بالحقول التابع للجدول الثاني ويتم ادراج null في قيم الحقول التابع للجدول الثاني في السجلات التي لا يوجد لها قيم مطابقة

وجميع قيم الحقول التابع للجدول الثاني والقيم المطابقة لها الخاصة بالحقول التابع للجدول الاول ويتم ادراج null في قيم الحقول التابع للجدول الاول في السجلات التي لا يوجد لها قيم مطابقة

وتأخذ الصيغة العامة:

```
SELECT table1.column1,table1.column2,table2.column1,....
```

```
FROM table1 FULL JOIN table2
```

```
ON table1.matching_column = table2.matching_column;
```

Full join

```
SELECT Student.NAME, StudentCourse.COURSE_ID  
FROM Student FULL JOIN StudentCourse  
ON StudentCourse.ROLL_NO = Student.ROLL_NO;
```

NAME	COURSE_ID
HARSH	1
PRATIK	2
RIYANKA	2
DEEP	3
SAPTARHI	1
DHANRAJ	NULL
ROHIT	NULL
NIRAJ	NULL
NULL	9
NULL	10
NULL	11

Natural join (using)

يستخدم في عملية الربط اعتمادا على حقول ذات اسماء مشتركة بين جدولين والتي تحتوي على قيم متطابقة ولا يكتب داخلها كلمة `on` لأنها تستخدم ضمنا في التعبير `natural join`

قواعد بيانات

الدرس 7

توليد وحذف قواعد البيانات

لتوليد قاعدة بيانات جديدة تستخدم الصيغة التالية:

```
CREATE DATABASE database_name;
```

لحذف قاعدة بيانات تستخدم الصيغة التالية:

```
Drop database database_name;
```

ملاحظة: استخدام صيغة create database مع اسم القاعدة يؤدي الى الغاء و استبدال تلك القاعدة

انشاء وحذف الجدول

لانشاء جدول يتم باستخدام مفتاح create table ويكتب بالصيغة العامة :

```
CREATE TABLE table_name (  
  column1 datatype,  
  column2 datatype,  
  column3 datatype,  
  ....  
);
```

لادراج عناصر داخل الجدول يستخدم مفتاح insert into

لحذف الجدول يستخدم مفتاح drop table *table_name*

لتفريغ سجلات الجدول يستخدم المفتاح truncate table *table_name*

ادراج جدول

مثال لعمل جدول و ادراج قيم داخلها

```
, CREATE TABLE Car (`code#` varchar(255), `CarName` varchar(255)  
(Description` varchar(255), `cost` varchar(255))`
```

لادراج القيم:

```
INSERT INTO car VALUES('BA11' , 'BMW' , '8 cylinder ,automatic',200)
```

code#	CarName	Description	cost
BA11	BMW	8 cylinder ,automatic	200

نسخ الجدول

لنسخ الجدول نستخدم الصيغة :

Create table table_name_copy **select * from** table_name

Example:

```
CREATE table `copied_T` select * from car
```

code#	CarName	Description	cost
BA11	BMW	8 cylinder ,automatic	200

نسخ الجدول

إذا اردنا نسخ الهيكل العام بدون السجلات نضيف الشرط `where 1=0`

Example :

```
CREATE table `copied_T2` select * from car where 1=0
```

```
code# CarName Description cost
```

تعديل بنية الجدول

يستخدم المفتاح `alter table` لتعديل الجدول اما باستخدامها لحذف او اضافة عمود (ADD, DROP COLUMN)

EXAMPLE:

```
ALTER TABLE `copied_T2` ADD COLUMN (`color` varchar(255));
```

code#	CarName	Description	cost	color
-------	---------	-------------	------	-------

```
ALTER TABLE `copied_T2` DROP COLUMN `color`
```

code#	CarName	Description	cost
-------	---------	-------------	------

القيود على الحقول

الاستخدام

يضمن عدم إمكانية احتواء العمود على قيمة فارغة

يعيّن قيمة افتراضية لعمود عندما لا يتم تحديد قيمة

يضمن أن جميع القيم الموجودة في عمود مختلفة

مزيج من NOT NULL و UNIQUE. يعرف كل صف في الجدول بشكل فريد

يعرف بشكل فريد صف / سجل في جدول آخر

يضمن أن جميع القيم الموجودة في عمود تفي بشرط معين تشبه عمل where

لتوليد قيم عددية بصورة آلية لحقل ما عند إضافة سجل جديد الى الجدول

القيود

Not null

default

unique

Primary key

Foreign key

Check

Identity-auto increment

```
Create table student (Fname varchar(20) Not null);
```

المنظار

المنظار هو جدول افتراضي يحتوي على معلومات كاي جدول من جداول قواعد البيانات ويساعد على استعادة البيانات ومنع المستخدمين من الوصول للجداول الاصلية ولا يمكن تعديل البيانات من خلال المنظار في العادة فهو نسخة قابلة للقراءة فقط

نستخدم الصيغة العامة:

```
Create view view_name as query;
```

تعديل البيانات في المنظار

شروط التعديل

1/ لا يحتوي على تابع تجميعي group by

2/ لا يحتوي على didtinct, top

3/ الا يحتوي الاستعلام على حقول بها عمليات حسابية

```
Alter view view_name as new_query;
```

اما حذف المنظار فأخذ الصيغة:

```
Drop view view_name;
```

الجدول المؤقتة

يمكننا عمل جداول مؤقتة لاداء هدف محدد ولفترة زمنية محددة باستخدام الصيغة التالية:

```
Create table #table_name (field1 type(),...);
```

الفهارس

صممت الفهارس لزيادة سرعة العمليات و تحسين الاداء عبر توليد جداول بحث داخلية فعندما تتم فهرسة جدول حسب حقل معين تتم فهرسة سجلات هذا الجدول حسب قيمة هذا الحقل ونمطه ممايسمح بايجاد هذه القيمة سريعا ولكن يجب مراعاة ان عمليات التعديل و الحذف و كثرة الفهارس قد تسبب سلبا على الاداء.

انواع الفهارس:

1/ فهارس فريدة لا تسمح بتكرار قيم في حقل الفرس و لانشائها نستخدم:

```
Create unique index index_name on table_name (field1);
```

2/ فهارس غير فريدة تسمح بتكرار القيم و لانشائها نستخدم:

```
Create index index_name on table_name (field1);
```

لحذف الفهارس

لحذف فهرس نستخدم الصيغة التالية:

```
Drop index table.index;
```

```
Drop index index on table;
```

قواعد بيانات

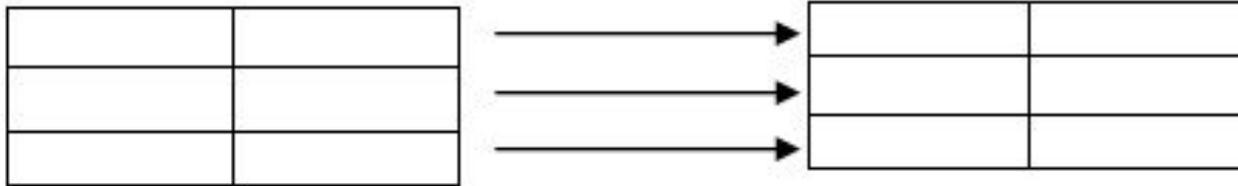
الدرس 8

التكامل المرجعي

التكامل المرجعي هو خاصية للبيانات تشير إلى أن جميع مراجعها صالحة فهو علاقة بين جدولين يحوي الجدول الأول على مفتاح أساسي و الجدول الثاني يرتبط مع الجدول الأول بمفتاح غريب له نفس اسم المفتاح الأساسي و نفس بياناته و نفس صفات

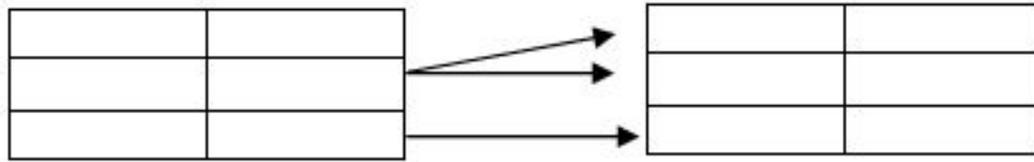
العلاقات التي تربط بين جدول و اخر (relationship):

1/ علاقة واحد لواحد: هذا النوع من العلاقات يتم بين جدولين على الأقل بحيث يكون سجل واحد فقط من الجدول الأول مرتبط مع سجل واحد فقط من الجدول الثاني و العكس صحيح

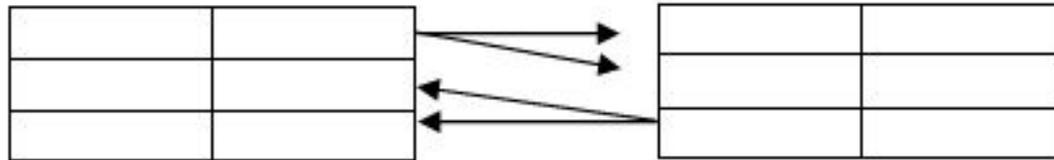


التكامل المرجعي

/2 هي علاقة رأس بأطراف: في هذا النوع تكون العلاقة بين الجدولين هي أن كل سجل من الجدول الأول له علاقة بسجل واحد على الأقل من الجدول الثاني بينما كل سجل من الجدول الثاني له علاقة بسجل واحد من الجدول الأول



/3 علاقة عديد لعديد : أي كل سجل من الجدول الأول يرتبط بعدة سجلات من الجدول الثاني و العكس صحيح أيضا



مثال

مثال

department_table

job_deprt	description
1	Accounting
2	Research
3	Field
4	Sales
5	Marketing
n	...

Primary Keys

employee_table

id	emp_fname	emp_lname	job_deprt
1	John	Carmack	3
2	Sylvester	Stallone	5
3	Arnold	Shazbaloney	2
4	Sonya	fromMK	1

Foreign Keys

المناقلات

هي وحدة عمل منطقية تتألف من سلسلة متتابعة من العمليات المنفصلة التي تحول النظام من وضعية استقرار الى وضعية استقرار اخرى حيث يعد ACID (الذرية والتماسك والعزل والاستمرارية) مجموعة من المبادئ التوجيهية التي يمكن أن تضمن إمكانية معالجة معاملات قاعدة البيانات بشكل موثوق. معاملة قاعدة البيانات هي أي عملية يتم إجراؤها داخل قاعدة بيانات ، مثل إنشاء سجل جديد أو تحديث البيانات داخل واحد.

يتم التعبير عن المناقلات باستخدام

```
Begin transaction transaction_name;
```

المناقلات

تعتبر نقاط الحفظ نقاط مرجعية او نقاط استرجاع ضمن المناقلة و يتم التعبير عنها ب `savepoint savepoint_name` ;
اما للرد او الغاء المناقلة او الرجوع لنقطة الحفظ يتم التعبير عنها باستخدام `rollback transaction`
عند اتمام المناقلة وتنفيذ الاوامر يتم تبليغ قاعدة البيانات بتثبيت التغييرات التي طرأت باستخدام التعبير
`commit transaction` ولكن بعد هذه المرحلة لا يمكن التراجع عن المناقلة

مثال صفحة 115

العزل

يستخدم العزل حتى لاتحدث مشاكل التضارب حي تحاول مناقلتان التفاعل مع نفس الاغراض في قاعدة البيانات المعقدة و تحاولان الوصول الى نفس البيانات في نفس الوقت وتختلف حدة المشكلة حسب العمليات التي تؤديها المناقلات

ولحل المشكلة تلجأ المناقلة الى طلب الملكية المؤقتة لقطعة من البيانات بوضع قفل عليها اذ ندعوا الية تحديد الاقفال على مصادر البيانات التي تعمل عليها المناقلة بدرجة العزل

وتصنف المناقلات الى نوعين : للقراءة و القراءة و الكتابة وللعزل اربع مستويات

1/ قراءة ملتزمة (read committed) الوضعية التلقائية وتتم غلق الموارد الى ان تتم المناقلة

2/ قراءة غير ملتزمة (read uncommitted) اسرع واكثر درجات العزل خطورة

3/ قراءة التسلسل (serializable) عدم قدرة أي مناقلة من تعديل سجل البيانات

4/قراءة متكررة (repeatable read) حل القراءات الخاطئة و القراءة المتكررة

مثال صفحة 118

الاجرائيات المخزنة

عبارة عن الية تلعب دور مشابه للدور الذي تلعبه المناظير ولكن بصورة موسعة, تقوم بمجموعة من الاعمال كاضافة سجل وتعديل سجلات

يتم استدعاء الاجرائيات المخزنة باستخدام تعبيرات execute,exec اما لانشاء اجرائية مخزنة create/replace procedure

لحذفها نستخدم drop procedure

لتعديلها alter procedure

انشاء و استخدام المتحولات

انواع المعاملات:

1/ معاملات الدخل تستخدم لتمرير معاملات الدخل

2/ معاملات الخرج لاعادة قيم من الاجرائية

3/ معاملات الدخل و الخرج لتامين الغرضين معا

المتحولات غرضها الاساسي تخزين قيم في الذاكرة يمكن استجاعتها باستخدام اسم المتحول لتصريحها

```
Declare var_name var_type (length)
```

لاسناد قيمة للمتحول بعد تصريحها نستخدم الصيغة

```
SET (select) @var_name = value;
```

التعابير الشرطية

1\ IF- ELSE:

IF condition

True code;

Else

False code;

ملاحظة : لكتابة أكثر من شرط يشترط كتابة begin , end

2\ case- when

Case expression

When value1 then result

.....

Else result;

End;

ص 147

حلقات التكرار

1\ while loop

While condition

Begin

Loop code

End;

مثال ص 149

المؤشرات

المؤشر عبارة عن طريقة لتمثيل مجموعة سجلات ضمن نص سيكول بحيث تسمح بالدوران على هذه السجلات و تؤثر على اداء قواعد البيانات

```
Declare cursorname CURSOR for ...;
```

```
Open cursorName;
```

```
Fetch cursorName..;
```

معالجة الخطأ

قد يتسبب بالخطأ استعمال خاطئ او عبارة سيكول خاطئة او تمرير بيانات غير مناسبة الى اجراء مخزنة..

ولا تتسبب اخطاء السيكول بالضرورة الى ايقاف تنفيذ النص البرمجي

تعبير RETURN تمكن الاجرائية المخزنة من ارجاع قيمة اذ تستخدم الصيغة التالية في تحديد اذا كان تنفيذ الاجراء ناجح ام لا
عادة تعيد القيمة صفر اذا كان ناجح

RETURN value;

المتحول العام @@error من اهم المتحولات في معالجة الاخطاء حيث ياخذ القيمة صفر في حال عدم ظهور أي خطأ

امن البيانات

كلما زادت اهمية البيانات زادت ضرورة الحفاظ عليها

عند تطبيق نظام امني لابد لنا من التفكير باجرائيين اساسيين

1/ التحقق من الهوية : باستخدام اسم دخول وكلمة المرور

2/ الصلاحيات و السماحيات : تحديد صلاحيات المستخدمين بعد التحقق من هويتهم

وتعطى الصلاحيات هذه من مدير النظام اذا يتم انشاء حساب لهذا المستخدم ثم توزيع الصلاحيات على المستخدمين

الادوار والمناظير في الامن

الادوار: هي مجموعة تضع لها صلاحيات محددة، ويضاف إليها المستخدمون، وهذا أسهل من إنشاء وتكرار صلاحيات لكل مستخدم، فأنت تنشئ الدور، وتضم إليها المستخدمين. من أمثلتها

`db_owner` مالك قاعدة المعطيات، السماح بإنشاء جداول وأغراض أخرى، ومعطيات أخرى.

`db_accessadmin` يمكن أن يدير ولوج تسجيلات الدخول إلى قاعدة المعطيات

`db_backupoperator` يمكن أن يقوم بنسخ احتياطي لقاعدة البيانات.

المناظير تستخدم لحجب بعض المعلومات واعطاء صلاحية التعامل مع الحقول المسموح له رؤيتها

التعبير Grant

يتم استخدامه لتحويل مستخدم بصلاحيات من و هناك نوعان من الصلاحيات:

1/ صلاحيات مستوى التصريح: صلاحيات تتضمن عمليات غير مرتبطة باغراض موجودة محددة في قاعدة البيانات

مثل create table, create procedure, create view, create database

GRANT prevelage_name TO user_name

2/ صلاحيات مستوى الاغراض: صلاحيات الوصول لاغراض محددة داخل قواعد البيانات مثل تحديد صلاحية الوصول الى الجداول: select, update, insert, delete, execute

GRANT prevelage_name ON resource TO user_name

التعبير revoke

يستخدم لإزالة أو حجب صلاحيات تم إعطاؤها بالتعبير grant أو حصل عليها بشكل تلقائي

```
REVOKE prevelage_name TO user_name
```

```
REVOKE prevelage_name ON resource TO user_name
```

القادحات

القادحات هي وسيلة لتأمين تطبيق قواعد العمل و تكامل البيانات و اتساقها و تماسكها داخل قاعدة البيانات فهي نوع خاص من الاجرائيات المخزنة المرتبطة بجدول معين يتم تنفيذة بصورة الية عند أي تعديل على هذا الجدول و نستطيع انشاؤ اكثر من قادح على جدول واحد و تنقسم لقسمين حسب عدد تنفيذ نص القادح

1/ قادحات على مستوى التعبير

2/ قادحات على مستوى السجل

لانشاء القادح نستخدم التعبير `CREATE TRIGGER trigger_name`

التوابع

لانشاء التوابع المعرفة من قبل المستخدم نستخدم التعبير

```
Create function function_name[[parameter_list]]
```

اما استدعاء التوابع بالتعبير:

```
Select function_name (parameter_list) as test;
```