جامعة طرابلس
كلية تقنية المعلومات

Advanced Databases قواعد البيانات المتقدمة
ITSE312

أستاذ المادة – حسن علي حسن

h.ebrahem@uot.edu.ly

المحاضرة الحادية عشر– أمن قاعدة البيانات والتحكم في الوصول

Database Security and Access Control

مواضيع المحاضرة الحادية عشر

- إنشاء مستخدم Create User
- حذف مستخدم DROP User
- تغيير كلمة السر للمستخدمين Changing Password for Accounts
- الامتيازات Privileges
  - جملة منح الامتياز GRANT Statement
  - جملة إلغاء الامتياز REVOKE Statement

Database Security أمن قاعدة البيانات

- نظرا لخطورة وأهمية البيانات لدى جميع المؤسسات والمنظمات والشركات، تعتبر البيانات ذات قيمة عالية، بالتالي يجب المحافظة عليها من الاختراق والتلاعب بتوفير أمن Security وترخيص Authorization للوصول إليها والاستفادة منها، وفي نفس الوقت يجب أن تكون البيانات متاحة للمستخدمين المخولين الذين لديهم امتيازات بالتعامل معها.

Database Security أمن قاعدة البيانات

- فقدان الأمن يؤدي إلى مجموعة من التهديدات تتعلق بفقدان **السرية Secrecy**، **السلامة Integrity**، **والتوفر Availability** كالتالي:

1. **السرية Secrecy**: تشير إلى حماية البيانات من المستخدمين الغير مصرح لهم، أي يجب أن يكونوا غير قادرين على رؤية الأشياء الغير مخولين برؤيتها.
2. **السلامة Integrity**: تشير إلى حماية البيانات من التحديث الغير مصرح به من (إدخال، حذف، أو تعديل)، بمعنى المستخدمون يجب أن يكونوا غير قادرين على تعديل الأشياء الغير مخولين بتعديلها.
3. **التوفر Availability**: تشير إلى توفر البيانات للمستخدمين المصرح لهم، المستخدمون يجب أن يكونوا قادرين على رؤية الأشياء المسموح لهم برؤيتها.

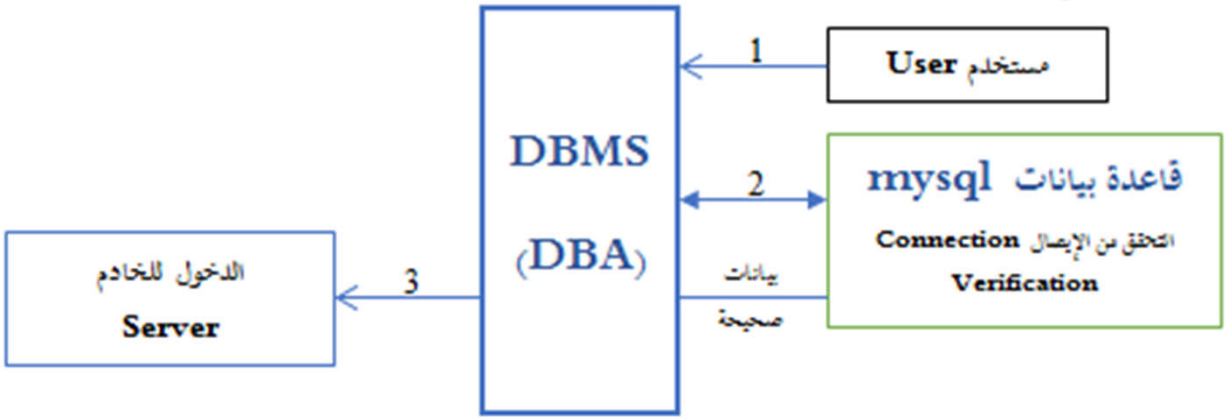
Database Security أمن قاعدة البيانات

- مسؤول قاعدة البيانات Database Administrator (DBA) هو المسؤول عن إدارة نظام قاعدة البيانات ولديه جميع الصلاحيات (الامتيازات) ويسمى Root أو Superuser. يقوم DBA بإنشاء حساب للمستخدمين ومنحهم الامتيازات أو إلغاء منهم الامتيازات الممنوحة لهم بالوصول لقاعدة البيانات، منح الصلاحيات للمستخدمين يعتمد على سياسة المؤسسة.
- يطبق DBMS نظام سيطرة متطور للتحكم بالدخول والذي يسمح بالوصول الشامل لقاعدة البيانات Database ويتحكم في عمليات المستخدم User، حيث يمنع المستخدمين الغير المخولين من الدخول لنظام قاعدة البيانات Database.

Database Security أمن قاعدة البيانات

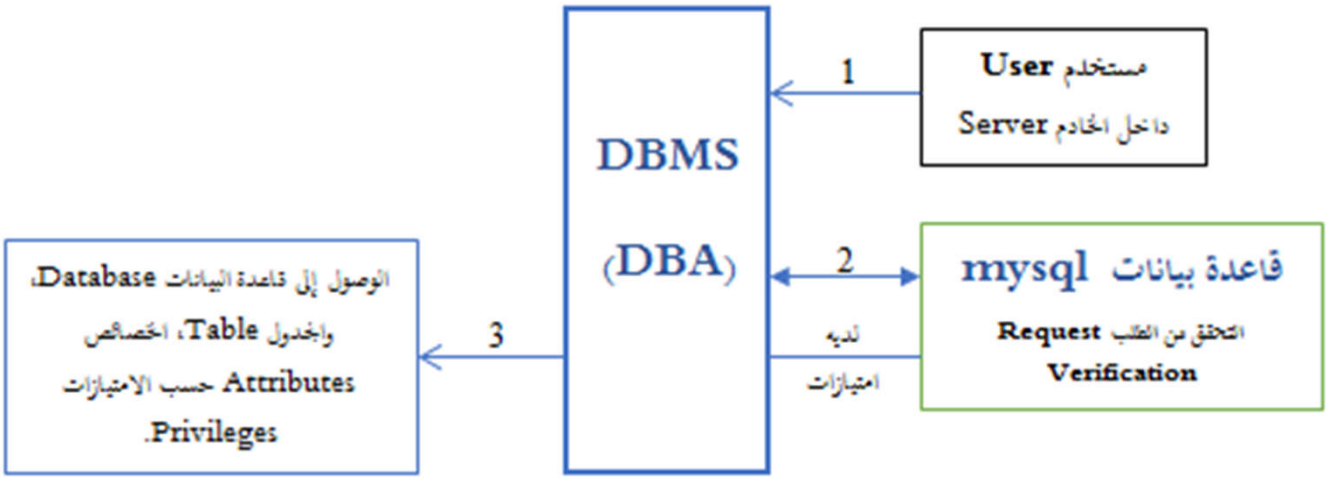
- عندما يحاول المستخدم الوصول إلى الخادم Server عن طريق DBMS تمر بمرحلتين Two Stages:

1- التحقق من الاتصال Connection Verification: عندما يحاول المستخدم الوصول إلى خادم قاعدة بيانات MySQL Server، يتم التحقق من اسم المستخدم Username وكلمة سر Password، مع التأكد هل حساب المستخدم مقفل Lockedأم لا Unlocked. الشكل التالي يوضح خطوات التحقق من الدخول للخادم.



Database Security أمن قاعدة البيانات

2- **التحقق من الطلب Request Verification**: بعدما يتم تأسيس اتصال ناجح بالخادم، ويتم دخول المستخدم للخادم، فإن لكل جملة Statement يتم اجرائها من قبل المستخدم يقوم الخادم بالتحقق من أن المستخدم لديه امتيازات Privileges كافية لتنفيذ الجملة أم لا. أنظر الشكل التالي



قاعدة البيانات mysql

عندما يتم تحميل (تنصيب) Install نظام إدارة قواعد البيانات MySQL يتم تكوين قاعدة بيانات باسم **mysql**، يتم إنشائها بشكل آلي. تحتوي قاعدة بيانات mysql على عدة جداول، سيتم التركيز على **خمس جداول رئيسية** تسمى جداول المنح **Grant Tables**، كل جدول يمتلك خصائص مختلفة تحدد إمكانية وصول المستخدم لقواعد البيانات والجداول والخصائص التي بداخلها.

يتم التعامل مع هذه الجداول داخل قاعدة البيانات mysql في أغلب الأحيان بشكل غير مباشر عند استخدام أمر المنح **GRANT** أو الإلغاء (المنع) **REVOKE**، بمعنى عند تنفيذ أمر المنح أو المنع لا يتم تحديد الجداول في الأمر مباشرة، وإنما يقوم الخادم بالتعامل مع هذه الجداول.

جدول User

- جدول User يحتوي على حسابات المستخدمين. الشكل يبين تركيبة جدول User، مسؤول قاعدة البيانات DBA يستخدم الخصائص الثلاثة الأولى (Host، User، Password) في قبول أو رفض المستخدم من الدخول للخادم Server الخاص بقاعدة البيانات، وذلك بالتحقق من المضيف والاسم وكلمة السر. بعض الخصائص الأخرى تحدد الامتيازات الممنوحة للمستخدم. أي امتياز يتم منحه في هذا الجدول يسمى امتياز عام Global privilege، لأنه يسمح بالوصول إلى جميع قواعد البيانات الموجودة على الخادم.

```
Create table user ( Host  varchar (60) binary NOT NULL default ' ',
User  varchar (16) binary NOT NULL default ' ',
Password  varchar (45) binary NOT NULL default ' ',
Select_priv  enum ('N' ,'Y') NOT NULL default 'N',
Insert_priv  enum ('N' ,'Y') NOT NULL default 'N',
Update_priv  enum ('N' ,'Y') NOT NULL default 'N',
Delete_priv  enum ('N' ,'Y') NOT NULL default ' N ',
Create_priv  enum ('N' ,'Y') NOT NULL default 'N',
Drop_priv  enum ('N' ,'Y') NOT NULL default 'N',
```

جدول Db

- جدول Db يحتوي على الامتيازات الممنوحة للمستخدمين لقاعدة البيانات معينة ومن أي مضيف Host. يستخدم هذا الجدول لتحديد أي قاعدة بيانات يستطيع المستخدم أن يتعامل معها ومن أي مضيف Host، وكذلك الأوامر التي يمكن تنفيذها على قواعد البيانات. الامتياز يُمنح لقاعدة بيانات محددة في جدول Db ويطبق على كل الكائنات في قاعدة البيانات مثل، الإجراءات المخزنة Stored Procedures والقوادح Triggers والمناظير Views والجداول Tables. أنظر الشكل.

```
Create table Db ( Host char (60) binary NOT NULL default ' ',
Db char (64) binary NOT NULL default ' ',
User char (16) binary NOT NULL default ' ',
Select_priv enum ('N','Y') NOT NULL default 'N',
Insert_priv enum ('N','Y') NOT NULL default 'N',
Update_priv enum ('N','Y') NOT NULL default 'N',
Delete_priv enum ('N','Y') NOT NULL default 'N',
```

الجدول Host

- جدول Host يستخدم مع جدول Db لتحديد أي قاعدة بيانات تكون متاحة لعدة مضيفين Hosts. أنظر الشكل.
- على سبيل المثال، لمنح مستخدم الإذن بالوصول لقاعدة بيانات معينة من عدة مضيفين، يتم في سجل المستخدم داخل جدول Db ترك خاصية المضيف Host فارغة، ثم يتم في جدول المضيف Host، إنشاء عدة سجلات تخص المستخدم، قيمة كل خاصية Hostفي هذه السجلات تخص مضيف معين.

```
Create table Host (Host  char (60) binary NOT NULL default '',
Db  char (64) binary NOT NULL default '',
Select_priv  enum ('N','Y') NOT NULL default 'N',
Insert_priv  enum ('N','Y') NOT NULL default 'N',
Update_priv  enum ('N','Y') NOT NULL default 'N',
Delete_priv  enum ('N','Y') NOT NULL default 'N',
Create_priv  enum ('N','Y') NOT NULL default 'N',
```

الجدول Tables_priv

- يحتوي هذ الجدول على الامتيازات الممنوحة للمستخدم والخاصة بجدول معين. أي امتياز يمنح في خصائص هذا الجدول Tables_priv يسمح للمستخدم بالوصول لجدول محدد وجميع الخصائص التي بداخله. أنظر الشكل

```
Create Table Tables_priv ( Host  char (60) binary NOT NULL default '',
Db char (64) binary NOT NULL default '',
User char (16) binary NOT NULL default '',
Table_name  char (60) binary NOT NULL default '',
```

Columns_priv الجدول

- الامتياز الذي يتم منحه في هذا الجدول يطبق على خاصية محدد فقط في الجدول، بمعنى يسمح للمستخدم بالوصول إلى خاصية معينة فقط داخل الجدول، أنظر الشكل.

```
Create Table Columns_priv ( Host  char (60) binary NOT NULL default '',
Db char (64) binary NOT NULL default '',
User char (16) binary NOT NULL default '',
Table_name  char (64) binary NOT NULL default '',
```

خادم قاعدة البيانات MySQL

- يقوم خادم قاعدة البيانات MySQL بقراءة جداول المنح ووضع محتوياتها في الذاكرة عند بدء تشغيله Startup، ويتم الرجوع إليها عندما يقوم المستخدم بتنفيذ أي عملية للتأكد من الامتياز الممنوح للمستخدم. عند القيام بالتعديل في الامتيازات الممنوحة للمستخدمين داخل الجداول، يتم إعلام الخادم بإعادة تحميل Reload المحتويات في الذاكرة من جديد.
- بمجرد دخول المستخدم للخادم يتم فحص الامتيازات الممنوحة له، عند قيامه بعملية على قاعدة البيانات يبدأ خادم قاعدة البيانات بالتحقق من الامتيازات الموجودة في الجداول الثلاثة الاولى (User, Db, Host). في حالة ما لم يتم توفر الامتيازات بها يتم استخدام الجدولين Table_priv وColumn_priv.

إنشاء مستخدم Create User

- في نظام DBMS تستطيع تحديد من المستخدم المسموح له بالدخول إلى خادم قاعدة البيانات وكذلك من أي مكان. وبالتالي فإن حساب الدخول Account في MySQL يتكون من اسم المستخدم Username واسم المضيف Host Name ويفصل بينهما بعلامة (آت @).
- **على سبيل المثال**، إذا أراد المستخدم root الاتصال من المضيف myhost.org بخادم قاعدة البيانات فإن اسم حساب الدخول يكون root@myhost.org، تستطيع تكوين عدة حسابات Accounts بنفس الاسم ولكن من عدة مضيفين مختلفين Hosts ولديهم امتيازات (صلاحيات) مختلفة. إن اسم المستخدم Username والمضيف Host يتم تخزينهم في جدول **User**.

إنشاء مستخدم بجملة CREATE USER

- يتم إنشاء مستخدم للدخول للنظام بإحدى الطريقتين باستخدام جملة Create أو Insert كالتالي:
- **إنشاء مستخدم بجملة CREATE USER**
- تستخدم جملة CREATE USER لإنشاء مستخدم جديد داخل قاعدة بيانات Database، والصيغة العامة لهذا الامر كما في الشكل.

CREATE USER اسم المستخدم IDENTIFIED BY كلمة السر ;

- نلاحظ من الشكل، يتم انشاء مستخدم وتخصيص كلمة سر للدخول. اسم المستخدم يكون بالصيغة التالية 'username'@'hostname'، يتم تحديد كلمة السر بعد جملة IDENTIFIED BY، كلمة السر يتم كتابيها بنص عادي، يقوم DBMS بتشفير كلمة السر بشكل آلي عند تخزينها في جدول User.

مثال – إنشاء مستخدم بجملة CREATE USER

- **مثال:** إنشاء مستخدم جديد باسم dbadmin يتصل من المضيف localhost بكلمة السر CrEate-User، يكون الأمر بالصيغة كما في الشكل.

```
CREATE USER 'dbadmin'@'localhost' IDENTIFIED BY 'CrEate-User';
```

- **مثال:** إنشاء مستخدم جديد باسم superadmin يستطيع الوصول من أي مضيف Host. لتحديد إمكانية الوصول من أي مضيف نستخدم علامة النسبة المئوية % بدل كتابة اسم المضيف Host، أنظر الشكل.

```
CREATE USER 'superadmin'@'%' IDENTIFIED BY 'Secured' ;
```

إنشاء مستخدم بجملة INSERT

- تستخدم جملة INSERT لإنشاء مستخدم جديد، تقوم هذه الجملة بإدخال سجل جديد في جدول User. عند استخدام هذا الأمر يجب استخدام الدالة PASSWORD لتشفير كلمة السر أثناء إدخال السجل في جدول User. تقوم الدالة PASSWORD بتشفير كلمة السر لكي لا تكون مرئية عند عرض السجل من الجدول باستخدام جملة SELECT.
- مثال: إنشاء مستخدم جديد باستخدام جملة INSERT، أنظر الشكل.

```
INSERT INTO User (host, user, password) VALUES ('localhost' , 'dbadmin' ,
PASSWORD('CrEate-User') ) ;
```

- تنبيه: أمر CREATE أو INSERT يتطلب توفر امتياز للمستخدم لكي يستطيع استعمال هذه الأوامر.

حذف مستخدم DROP User

- أحيانا بعد انشاء مستخدم قد تحتاج لإلغاء (حذف) هذا المستخدم من النظام.
- إذا أردنا حذف المستخدم من النظام، نستخدم الصيغة العامة في الشكل.

```
DROP USER 'username'@'hostname' ;
```

- مثال: حذف المستخدم superadminالمتصل من أي مضيف، نستخدم الامر في الشكل.

```
Mysql> DROP USER 'superadmin'@'%' ;
```

الاتصال بخادم MySQL

- عند محاولة الاتصال بخادم MySQL يتم تحدد مجموعة من المعاملات Parameters والتي تتمثل في اسم الخادم واسم المستخدم وكلمة المرور.
- توجد بعض البرامج توفر خدمة الخادم MySQL Server محلي Local في نفس الجهاز، يمكن الاستعانة بها وتشغيلها عن طريق سطر الأوامر Command-line Prompt، على سبيل المثال، إذا كان المستخدم dbadmin يريد الدخول من المضيف Localhostبكلمة سر مشفرة، يتم كما في الشكل.

```
[root @ host] $ mysql –h localhost –u dbadmin –p

Enter Password: ******
Welcome to the Mysql monitor. Commands end with ; or \g.
Mysql>exit;
[root @ host] $ mysql –u dbadmin@myhost.org –p
Enter Password: ******
Welcome to the Mysql monitor. Commands end with ; or \g.
Mysql>exit;
[root @ host] $ mysql --host=localhost --user=dbadmin --password=
```

تغيير كلمة السر لحسابات المستخدمين
Changing Password for Accounts

- يمتلك نظام DBMS عدة جمل Statements تستخدم لتغيير **كلمة السر** PASSWORD لحساب Account المستخدمين، هذه الجمل المستخدمة في تغيير كلمة السر تتمثل في UPDATE Statement, SET PASSWORD Statement and GRANT USAGE Statement.
- توجد بعض التنبيهات الواجب الاهتمام بها قبل البدء في تغيير كلمة السر Password الخاص بالمستخدم وهي:
- تحديد أي مستخدم تريد تغيير كلمة السر الخاص به.
- من أي مضيف Host المستخدم متصل، المراد تغيير له كلمة السر.
- تغيير كلمة السر بدون تخطيط مسبق، يؤدي إلى فقد بعض التطبيقات الوصول إلى قاعدة البيانات.

تغيير كلمة السر Password باستخدام جملة التحديث UPDATE

- الطريق الأول لتغيير كلمة السر هي استخدام جملة التعديل UPDATE Statement، والتي تقوم بتعديل كلمة السر للخاصية userالتي تحتوي على اسم المستخدم، والخاصية hostالتي بها اسم المضيف الخاص بالمستخدم، هذه الخصائص موجودة داخل جدول USER في قاعدة البيانات mysql. الصيغة العامة كما في الشكل.

```
UPDATE USER SET password = PASSWORD(كلمة السر الجديدة) WHERE user = اسم
المستخدم AND host = اسم المضيف ;
```

- من الشكل نلاحظ وجود الدالة PASSWORD والتي تقوم بتشفير كلمة السر عند التخزين.
- بعد تنفيذ جملة التحديث UPDATEفي الشكل، تحتاج دائما لتنفيذ جملة FLUSH PRIVILEGES لإعادة تحميل التعديل من جديد في الذاكرة.

تغيير كلمة السر Password باستخدام جملة التحديث UPDATE

- **مثال:** بفرض أننا نريد تغيير كلمة السر Password بكلمة السر الجديدة Secret1974 للمستخدم myhost المتصل من المضيف myhost.org، نحتاج لتنفيذ الامر في الشكل.

```
Mysql>UPDATE user SET password = PASSWORD('Secret1974') WHERE user
=' myhost' AND host = 'myhost.org' ;
Mysql> FLUSH PRIVILEGES ;
```

Privileges الامتيازات

- الامتيازات هي العمليات التي يُسمح للمستخدم بتنفيذها على جداول قواعد البيانات، الجدول يبين مجموعة من الامتيازات (الصلاحيات) التي تمنح للمستخدم. تستخدم هذه الامتيازات في تمكين المستخدم من تنفيذ مجموعة من العمليات على قواعد البيانات والجداول التي بداخلها، يتم ذلك عن طريق خادم MySQL.

الصلاحية أو الامتياز	الاستخدام
ALL [PRIVILEGES]	منح جميع الصلاحيات للمستخدم، عدا صلاحية Grant Option.
ALTER	منح صلاحية التعديل في بنية الجدول Alter Table.
CREATE	منح صلاحية إنشاء قاعدة بيانات Database والجداول Tables والفهارس Indexes
CREATE USER	منح صلاحية إنشاء مستخدم CREATE USER وحذف مستخدم DROP USER وتغيير اسم المستخدم RENAME USER وإلغاء كافة الامتيازات أو الصلاحيات REVOKE ALL PRIVILEGES.
CREATE VIEW	منح صلاحية إنشاء وتعديل منظار View.
DELETE	منح صلاحية حذف سجلات Delete من الجدول.
DROP	منح صلاحية حذف قاعدة بيانات Database وجدول Table ومنظار View.
GRANT OPTION	منح أو إلغاء صلاحيات (امتيازات) من أو إلى مستخدمين أخرين.
INDEX	منح صلاحية إنشاء فهرسة Index.
INSERT	منح صلاحية جملة الادخال Insert.
SELECT	منح صلاحية جملة الاستفسار Select.
SHOW VIEW	منح صلاحية جملة عرض تركيبة المنظار SHOW CREATE VIEW
TRIGGER	منح صلاحية استخدام القادح Trigger.
UPDATE	منح صلاحية جملة التعديل Update.

GRANT Statement جملة المنح

- تتم إدارة الأمان على قاعدة البيانات عن طريق الخادم من خلال جملة منح الامتياز GRANT وجملة إلغاء الامتياز REVOKE.
- **GRANT Statement جملة المنح**
- يزود نظام DBMA جملة منح الامتياز GRANT والتي تسمح بمنح امتيازات للمستخدم للوصول إلى جداول قاعدة البيانات والقيام بمجموعة من العمليات عليها. الصيغة العامة في الشكل.

```
GRANT Privilege (اسم الخاصية , اسم الخاصية, ...), Privilege (اسم الخاصية, ...) ... ON اسم قاعدة
اسم الجدول.البيانات | *.* TO اسم المستخدم IDENTIFIED BY 'كلمةالسر', ... WITH
GRANT OPTION ;
```

أمثلة – جملة المنح GRANT Statement

- **مثال:** إنشاء مستخدم (حساب) باسم super من المضيف localhost يمتلك جميع الصلاحيات على جميع قواعد البيانات ويقوم بمنح الامتيازات الممنوحة له إلى المستخدمين الآخرين، كما في الشكل.

```
Mysql>CREATE USER 'super'@'localhost' IDENTIFIED BY 'SecurePass1';
Mysql>GRANT ALL ON *.* TO 'super'@'localhost' WITH GRANT
OPTION;
```

- من الشكل السابق نلاحظ، في الأمر الأول، تم إنشاء مستخدم باسم super من المضيف localhost، أما في الأمر الثاني تم منح المستخدم جميع الصلاحيات *.* على كافة قواعد البيانات databases وكل الكائنات objects في قواعد البيانات. المستخدم super لديه صلاحية منح جميع صلاحياته إلى مستخدم أخر باستخدام الامر GRANT OPTION.

أمثلة – جملة المنح GRANT Statement

- **مثال**: إنشاء مستخدم super2 لديه الوصول الكامل لقاعدة البيانات **كلية_التقنية** من أي host بكلمة سر SecurePass2 مع إمكانية منح الصلاحيات للأخرين، أنظر الشكل.

```
Mysql>CREATE USER 'super2'@'%' IDENTIFIED BY 'SecurePass2' ;
Mysql>GRANT ALL ON كلية_التقنية.* TO 'super2'@'%' WITH GRANT
OPTION ;
```

- مثال: إنشاء مستخدم Ali لديه صلاحية التعديل UPDATEوالاسترجاع SELECT والإدخال INSERT على قاعدة البيانات كلية_التقنية ومن أي مضيف Host، أنظر الشكل.

```
Mysql>CREATE USER 'Ali'@'%' IDENTIFIED BY 'SecurePass3';
Mysql>GRANT SELECT, UPDATE, DELETE ON كلية_التقنية.* TO 'Ali'@'%';
```

أمثلة – جملة المنح GRANT Statement

- **مثال:** منح المستخدم السابق Aliامتياز التعديل UPDATE على خاصية الإيجار_الشهري في جدول الملكية، أنظر الشكل.

```
Mysql>GRANT  UPDATE(الإيجار_الشهري) ON الملكية.كلية_التقنية  TO 'Ali'@'%';
```

- **مثال:** لعرض الامتيازات الممنوحة للمستخدم السابق Aliنستخدم الأمر في الشكل التالي.

```
Mysql>SHOW GRANTS FOR 'Ali'@'%' ;
```

- عند تنفيذ الأمر في الشكل السابق، تظهر الامتيازات الممنوحة للمستخدم Aliكما في الشكل التالي.

```
GRANT SELECT, UPDATE, DELETE, UPDATE(الإيجار_الشهري) ON كلية_التقنية.* TO
'Ali'@'%'  IDENTIFIED  BY  PASSWORD
'5CBB3084DA19DDD293EEAA71A87B2A38C4048A2C'
```

أمثلة – جملة المنح GRANT Statement

- **مثال:** إنشاء مستخدم جديد باسم ahmed من المضيف localhoat بدون منحه امتيازات كما في الشكل

```
Mysql>CREATE USER 'Ahmed'@'localhoat' IDENTIFIED BY 'SecureP3';
```

- بفرض لدينا المستخدم أحمد لديه جميع الامتيازات هو المسؤول Rootعلى النظام، وتوجد قاعدة البيانات داخل الخادم تسمى **العقارات** وتحتوي الجداول التالية:
- **جدول الزبون (رقم الزبون، اسم الزبون)**
- **جدول التأجير (رقم الزبون، رقم الملكية، تاريخ التأجير، تاريخ النهاية).**
- **جدول الملكية (رقم الملكية، عنوان الملكية، الإيجار الشهري، رقم مالك العقار)**
- **جدول مالك العقار (رقم مالك العقار، اسم مالك العقار)**

أمثلة – جملة المنح GRANT Statement

- قام المستخدم أحمد بمنح بعض الامتيازات التي لديه للمستخدمين أنور وفرح، كما في الشكل

```
Mysql>GRANT SELECT, INSERT ON الزبون.العقارات TO أنور@yah.com ,
فرح@yah.com WITH GRANT OPTION;
Mysql>GRANT SELECT ON التأجير.العقارات TO أنور@yah.com, فرح@yah.com
WITH GRANT OPTION;
```

- قام المستخدم فرح بمنح الامتيازات التي لديه للمستخدم نور بدون خيار GRANT OPTION، أي لا يستطيع المستخدم نور منح امتيازاته لمستخدمين آخرين، كما في الشكل

```
Mysql>GRANT SELECT, INSERT ON الزبون.العقارات TO نور@yah.com ;
Mysql>GRANT SELECT ON التأجير.العقارات TO نور@yah.com ;
```

أمثلة – جملة المنح GRANT Statement

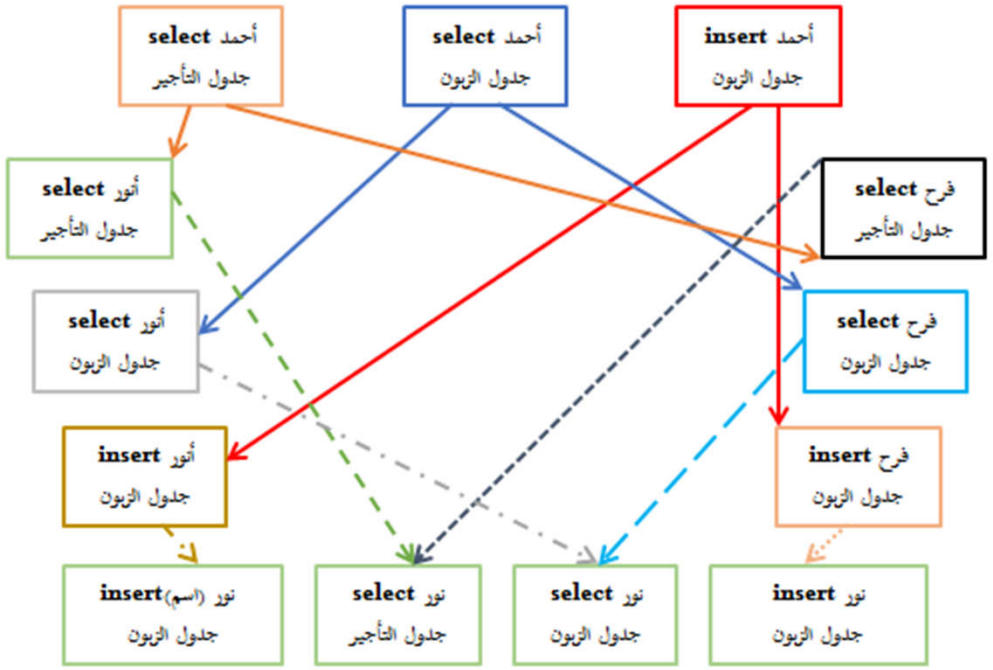
- قام المستخدم أنور بمنح الامتيازات التي لديه للمستخدم نور بدون خيار GRANT OPTION، كما في الشكل

```
Mysql>GRANT SELECT, INSERT(اسم_الزبون) ON الزبون.العقارات TO نور@yah.com ;
Mysql>GRANT SELECT ON التأجير.العقارات TO نور@yah.com ;
```

- نلاحظ أن المستخدم نور لديه صلاحيات مزدوجة، أي لديه امتيازات من المستخدم أنور ومن المستخدم فرح، نوضح التداخل السابق في الامتيازات بين مدير النظام أحمد والمستخدمين (أنور، فرح، نور) بمخطط يسمى مخططات المنح **Grant Diagrams**.
- نظام SQL يستخدم هذا المخطط لمتابعة الامتيازات بين المستخدمين.

أمثلة – جملة المنح GRANT Statement

- مخططات المنح **Grant Diagrams** كما في الشكل.



- على سبيل المثال من الشكل، نلاحظ أن أحمد المسؤول على قاعدة البيانات منح ثلاثة امتيازات إلى المستخدم فرح والمستخدم أنور. قام المستخدم أنور بمنح امتيازاته إلى المستخدم نور وقام المستخدم فرح بمنح امتيازاته للمستخدم نور، وبالتالي نلاحظ المستخدم نور لديه نفس الامتياز (جملة selectعلى جدول التأجير، جملة select على جدول الزبون) من المستخدمين أنور وفرح.

REVOKE Statement جملة إلغاء

- في بعض الحالات تحتاج إلى إلغاء عدة امتيازات من بعض المستخدمين، على سبيل المثال، تحتاج لإلغاء امتياز SELECT من مستخدم ومنحه لمستخدم آخر، وأحيانا نجد بعض المستخدمين قاموا بمنح امتيازاتهم لمستخدمين آخرين، لإتمام إلغاء الامتيازات يتم استخدام أمر REVOKE.
- يزود نظام DBMA جملة إلغاء (المنع) **REVOKE** الامتياز التي تسمح بإلغاء امتيازات المستخدم من الوصول إلى جداول قاعدة البيانات Database. الصيغة العامة كما في الشكل

```
REVOKE ALL PRIVILEGES | الامتياز [(اسم الخاصية)] , الامتياز [(اسم الخاصية)]... , GRANT
OPTION ON اسم الجدول.قاعدة البيانات | *.* FROM اسم المستخدم@اسم المضيف, ... RESTRICT
| CASCADE ;
```

أمثلة – جملة إلغاء REVOKE Statement

- **مثال:** لدينا المستخدم Ali يتصل من أي مضيف ولديه الامتيازات (الاختيار SELECT، التعديل UPDATE والحذف DELETE والتعديل على الخاصية (الإيجار_الشهري)UPDATE) لكل جداول قاعدة بيانات **العقارات**، أنظر الشكل

```
Mysql>GRANT SELECT, UPDATE, DELETE, UPDATE(الإيجار_الشهري) ON
العقارات.*TO 'Ali'@'%' ;
```

- عند تنفيذ امر عرض الامتيازات SHOW GRANTS FOR للمستخدم Ali، تظهر لنا الامتيازات الممنوحة للمستخدم Ali، كما في الشكل

```
GRANT SELECT, UPDATE, DELETE, UPDATE(الإيجار_الشهري) ON العقارات.* TO
'Ali'@'%' IDENTIFIED BY PASSWORD
'5CBB3084DA19DDD293EEAA71A87B2A38C4048A2C'
```

أمثلة – جملة إلغاء REVOKE Statement

- **مثال:** إلغاء الامتيازات UPDATE and DELETE الممنوحة للمستخدم Ali، يتم ذلك كما في الشكل

```
Mysql>REVOKE UPDATE, DELETE ON العقارات.* FROM 'Ali'@'%';
```

- عند تنفيذ أمر عرض امتيازات المستخدم Ali، تظهر لنا رسالة تبين الامتيازات الممنوحة المتبقية للمستخدم، كما في الشكل

```
GRANT SELECT, UPDATE(الإيجار_الشهري) ON العقارات.* TO 'Ali'@'%' IDENTIFIED BY
PASSWORD '5CBB3084DA19DDD293EEAA71A87B2A38C4048A2C'
```

أمثلة – جملة إلغاء REVOKE Statement

- يمكن إلغاء كل الامتيازات من مستخدم أو أكثر بالأمر، أنظر الشكل.

```
REVOKE ALL PRIVILEGES, GRANT OPTION FROM user [, user]...;
```

- لإلغاء كل الامتيازات السابقة للمستخدم Ali المتصل من أي مضيف، نستخدم الامر في الشكل

```
Mysql>REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'Ali'@'%';
```

- عند تنفيذ أمر عرض الامتيازات الممنوحة إلى حساب Aliفلن تجد أي امتيازات. تظهر رسالة عدم وجود أي امتيازات كما في الشكل

```
GRANT USAGE ON العقارات.* TO 'Ali'@'%' IDENTIFIED BY PASSWORD
'5CBB3084DA19DDD293EEAA71A87B2A38C4048A2C'
```

أمثلة – جملة إلغاء REVOKE Statement

- بالرجوع إلى مخططات المنح **Grant Diagrams** في الشكل السابق. نفرض أن المستخدم المسؤول على قاعدة البيانات (**أحمد**) قام بإبطال **REVOKE** بعض الامتيازات الممنوحة للمستخدم فرح كما في الشكل.

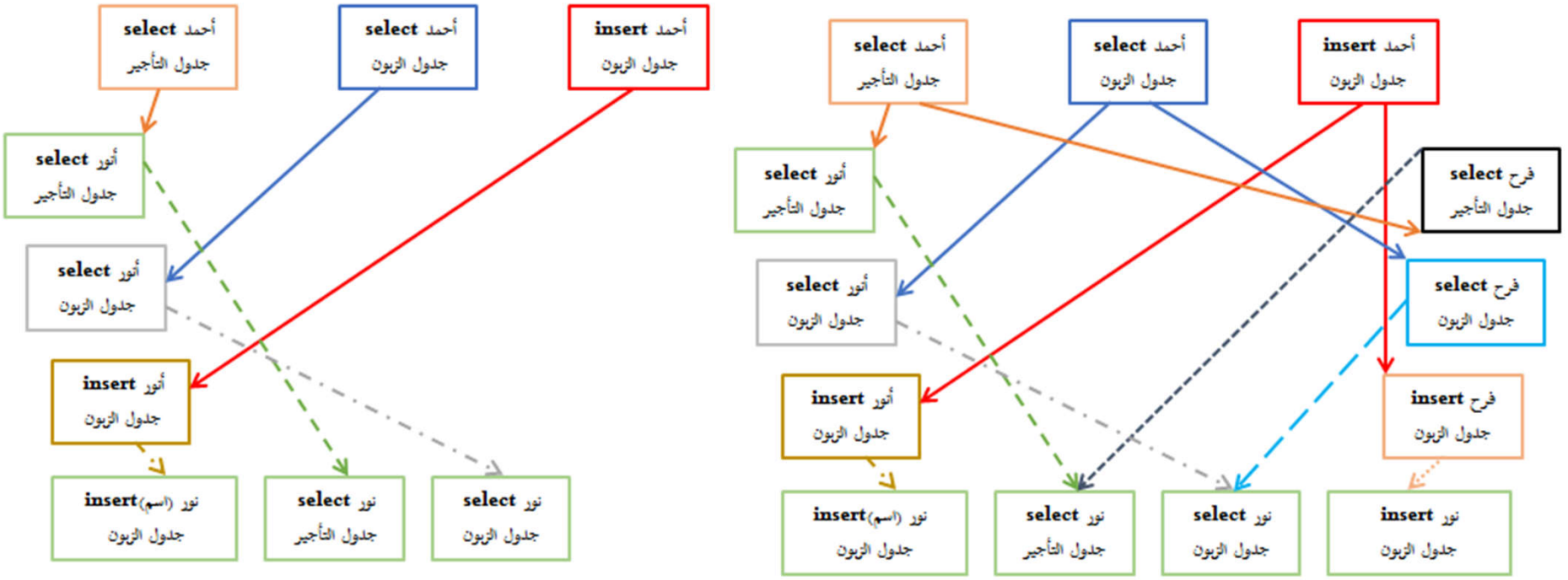
```
Mysql>REVOKE SELECT, INSERT ON الزبون.العقارات TO فرح@yah.com
CASCADE ;
Mysql>REVOKE SELECT ON التأجير.العقارات TO فرح@yah.com CASCADE ;
```

- نلاحظ من الشكل. عند قيام المسؤول على قاعدة البيانات بإلغاء الامتيازات الممنوحة للمستخدم فرح مع استخدام خيار CASCADE، يقوم مسؤول قاعدة البيانات DBA بحذف جميع الامتيازات الممنوحة للمستخدم فرح مع حذف الامتيازات التي منحها المستخدم فرح إلى المستخدم نور. بمعنى تتبع إلغاء الامتيازات الممنوحة من كل المستخدمين المرتبطين بالمستخدم فرح.

أمثلة – جملة إلغاء REVOKE Statement

```
Mysql>REVOKE SELECT, INSERT ON الزبون.العقارات TO فرح@yah.com
CASCADE ;
Mysql>REVOKE SELECT ON التأجير.العقارات TO فرح@yah.com CASCADE ;
```

- يكون شكل مخطط المنح Grant Diagram بعد قيام (أحمد) المسؤول على قاعدة البيانات بإلغاء الامتيازات من المستخدم (فرح) كما في الشكل على اليسار



أمثلة – جملة إلغاء REVOKE Statement

- كما نلاحظ بقاء الامتياز SELECT على (جدول الزبون وجدول التأجير) الممنوح للمستخدم نور، ذلك بسبب منح المستخدم أنور نفس الامتياز للمستخدم نور.
- هذا يعنى أن المستخدم نور مازال لديه امتيازات على قاعدة البيانات في الوقت الذي تم حذف امتيازاته من قبل مسؤول قاعدة البيانات باستخدام الخيار التتبع CASCADE.
- **تنبيه:** يجب الانتباه والحذر عند منح امتيازات لمستخدمين ومنحهم خيار GRANT OPTION، يترتب عليه أحيانا بقاء امكانية المستخدمين الوصول إلى جداول قاعدة البيانات.

Summary ملخص

- يحتفظ نظام إدارة قواعد البيانات MySQL بقاعدة بيانات باسم mysql بها عدة جداول تسمى جداول المنح Grant Tables، كل جدول يحدد امتيازات معينة ممنوحة للمستخدم، أي يتم عن طريقها تحديد إمكانية وصول المستخدم لقواعد البيانات والجداول والخصائص.
- قبل منح الصلاحيات يجب أن يتم إنشاء المستخدم داخل النظام، مع تحديد المضيف Host وكلمة السر الخاصة به. يتم إنشاء المستخدم باستخدام Create User أو جملة Insert. في حالة الحاجة لحذف Userمن النظام نستخدم Drop User.
- أحيانا نحتاج لتغيير كلمة السر الخاصة بالمستخدم، يتم ذلك عن طريق ثلاثة جمل UPDATE Statement, SET PASSWORD and GRANT USAGE Statement.
- بعد إنشاء المستخدم يتم منحه الامتيازات التي تتمثل في مجموعة من العمليات التي يُسمح للمستخدم بتنفيذها داخل قاعدة البيانات. يتم منح أو إلغاء الامتيازات بواسطة GRANT and REVOKE.

نهاية المحاضرة