UNIVERSITY of RWANDA

College of Science and Technology

School of ICT

Computer Science Department

Year 2

Lecturer: Muhayimana Odette
E-mail: muhayiodette06@gmail.com
Phone: 0788888571

CHAPIV: Structured Query Language (SQL) Programming

What is SQL?

SQL (*Structured Query Language*) is the international recognized standard language for database querying. The relational model of database was created by E.F Codd (Director of Research of IBM Centre of San José) in 1970. Many languages appeared after this creation:

•IBM Sequel (Structured English Query Language) in 1977 ,IBM Sequel/2 ,IBM System/R and IBM DB2

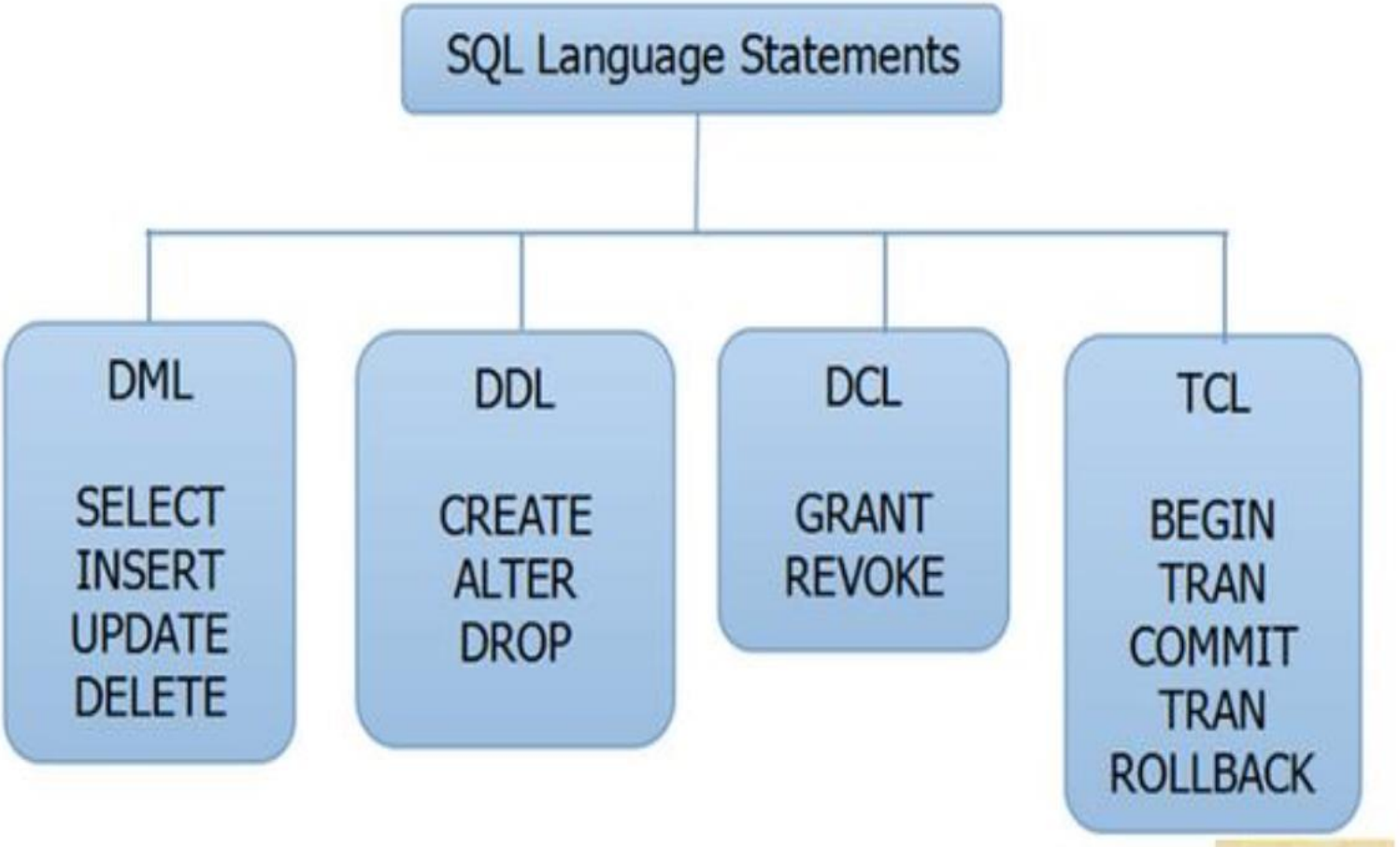
 These languages have enabled the creation of SQL. In 1986, SQL was normalized to SQL/86 by the ANSI; in 1989 the SQL/89 was approved. Since 1992, the norm of SQL is SQL/92 that is also called SQL 2.

However, there are some slight differences in SQL according to the RDBMS which is used. For example a code can run using Oracle, but not run using Access or SQL Server. In this case, you have to refer to the appropriate documentation of the RDBMS you are using.

Parts of SQL

- **SQL as a Data Definition Language (DDL)-** This part of SQL enables the creation of tables and relationship between those tables in a database.
- **SQL as a Data Manipulation Language (DML)-** This part of SQL enables to select, insert, update and delete data in a table in a database.
- **SQL as a Data Control Language (DCL)-**It's possible to grant or remove permissions to users in a database. This is made using the DCL.
- **Transactional SQL-** This part supports procedural programming and various functions for string processing, date processing, mathematics etc. The most popular transactional languages are **PL/SQL** (Procedural Language for SQL) of **Oracle** and **Transact-SQL or T-SQL** of **Microsoft SQL Server**.

Parts of SQL


SQL Language Statements
DML
SELECT
INSERT
UPDATE
DELETE
DDL
CREATE
ALTER
DROP
DCL
GRANT
REVOKE
TCL
BEGIN
TRAN
COMMIT
TRAN
ROLLBACK

Data Definition Language (DDL)

- Table Creation

The creation of tables is made using the statement CREATETABLE.

- Data types

For each column to be created, the data type should be précised. The following are data types :

Data Types	Syntax	Description
Alphanumeric type	CHAR(n)	Chain of characters with static length *n* (n<16383)
Alphanumeric type	VARCHAR(n)	Chain of characters with n maximum characters (n<16383)
Numeric Type	NUMBER(n)	Number with n characters
Numeric Type	SMALLINT	Signed integer of 16 bits (-32768 à 32757)

Data Definition Language (DDL)

- Table Creation

Numeric Type	INTEGER	Signed integer of 32 bits (-2E31 à 2E31-1)
Numeric Type	FLOAT	
Time type	DATE	
Time type	TIME	Hour with the form: 12:54:24.85
Time type	TIMESTAMP	Date and Hour

- Example of table creation

```
CREATETABLE STUDENTS
(
StudentID Varchar(5) not null,
Surname Varchar(50) not null,
FirstName Varchar(50),
BirthDate Date(10),
Faculty Varchar(40),
Department Varchar(40),
Primary Key (StudentID)
)
```

Data Definition Language (DDL)

- Table Deletion

The deletion of a table is made using the following syntax:

DROPTABLE table x

Example:
DROPTABLE CLIENTS

- Deleting a column of a table

It's made using the following syntax:

ALTERTABLE tablex
DROP COLUMN columnx

Example:

ALTERTABLE CLIENTS

DROP COLUMNAge

Data Definition Language (DDL)

Adding a column on a table

It's made using the following syntax:

ALTERTABLE tablex

ADD Column Columnx DataType

Example:

ALTERTABLE CLIENTS

ADD COLUMNTelephoneVarchar (10)

Data Manipulation Language (DML)

The SELECT Statement

The command SELECT is the most common used command in SQL.

It's for data extraction from the database and displaying to the interface. Its syntax is the following one:

SELECT [ALL] or [DISTINCT] <list of fields> or *(to select all fields) FROM <List of tables>[WHERE <logical condition>] There are

There are other clauses for the statement:

GROUP BY

HAVING

ORDER BY

Data Manipulation Language (DML)

Considering the following table CARS:

CARS

Make	Model	Series	CarID
Renault	18	RL	4698 SJ 45
Renault	Kangoo	RL	4568 HD 16
Renault	Kangoo	RL	6576 VE 38
Peugeot	106	KID	7845 ZS 83
Peugeot	309	Chorus	7647 ABY 82
Ford	Escort	Match	8562 EV 23

- The selection of all the columns of the table is written:

SELECT *

FROM CARS

The result will be:

Data Manipulation Language (DML)

CARS

Make	Model	Series	CarID
Renault	18	RL	4698 SJ 45
Renault	Kangoo	RL	4568 HD 16
Renault	Kangoo	RL	6576 VE 38
Peugeot	106	KID	7845 ZS 83
Peugeot	309	Chorus	7647 ABY 82
Ford	Escort	Match	8562 EV 23

- The selection of the columns Model and Series is written :

SELECT Model,Series

FROM CARS

The result will be:

Model	Series
18	RL
Kangoo	RL
Kangoo	RL
106	KID
309	Chorus
Escort	Match

Data Manipulation Language (DML)

Look at the column Model, **Kangoo** comes two times. The same with its Series **RL**. We do not need them to appear twice because it is a repetition.

To delete this redundancy we need to add the **option DISTINCT**.

•The selection of the columns Model and Serie without doubles is written :

SELECT DISTINCT Model, Series

FROM CARS

The result will be :

Model	Series
18	RL
Kangoo	RL
106	KID
309	Chorus
Escort	Match

Data Manipulation Language (DML)

- **Restriction or Selection**

The restriction or the selection consists of the selection of the rows which satisfied a logical condition. The restriction is introduced using the clause **WHERE** followed by the logical conditions instructed using logical operators.

- Simple Restrictions

Considering the following table:

CARS

Make	Model	Series	CarID	Meter
Renault	18	RL	4698 SJ 45	123450
Renault	Kangoo	RL	4568 HD 16	56000
Renault	Kangoo	RL	6576 VE 38	12000
Peugeot	106	KID	7845 ZS 83	75600
Peugeot	309	Chorus	7647 ABY 82	189500
Ford	Escort	Match	8562 EV 23	

Data Manipulation Language (DML)

- The list of cars with a meter less than 100,000 Km is given by:

SELECT *

FROM CARS

WHERE (Meter < 100000)

The result is:

Make	Model	Series	CarID	Meter
Renault	Kangoo	RL	4568 HD 16	56000
Renault	Kangoo	RL	6576 VE 38	12000
Peugeot	106	KID	7845 ZS 83	75600

- The list of cars (Make Meter) with a meter less than 100,000 Km is written:

SELECT Make,Meter

FROM CARS

WHERE (Meter< 100000)

*****This query consists of a projection and a selection**

The result is:

Data Manipulation Language (DML)

Make	Meter
Renault	56000
Renault	12000
Peugeot	75600

• The list of cars with a meter less or equal to 100,000 Km and higher or equal to 30,000 Km is written:

```
SELECT *
FROM CARS
WHERE (Meter >= 30000) AND (Meter <= 100000)
```

The result is:

Make	Model	Series	CarID	Meter
Renault	Kangoo	RL	4568 HD 16	56000
Peugeot	106	KID	7845 ZS 83	75600

Data Manipulation Language (DML)

- **Restrictions on Assemblage**

The SQL clauses **BETWEEN** and **IN** enable to make restrictions on assemblages.

•The list of cars with a meter less or equal to 100,000 Km and higher or equal to 30,000 Km (*written above)* can also be written:

SELECT *

FROM CARS

WHERE Meter BETWEEN 30000AND 100000

The result is:

Make	Model	Series	CarID	Meter
Renault	Kangoo	RL	4568 HD 16	56000
Peugeot	106	KID	7845 ZS 83	75600

Data Manipulation Language (DML)

• The list of cars which have a manufacturer Peugeot or Ford is :

SELECT *

FROM CARS

WHERE Make IN ("Peugeot","Ford")

The result is:

Make	Model	Series	CarID	Meter
Peugeot	106	KID	7845 ZS 83	75600
Peugeot	309	Chorus	7647 ABY 82	189500
Ford	Escort	Match	8562 EV 23	

Data Manipulation Language (DML)

- **Restriction on missing values**

• The list of cars with a missing meter information is written :

```
SELECT *
FROM CARS
WHERE Meter IS NULL
```

The result is:

Make	Model	Series	CarID	Meter
Ford	Escort	Match	8562 EV 23	

Data Manipulation Language (DML)

- **Sorting**

The SQL clause **ORDER BY** enables to sort values. The option **ASC** indicates that the sort is ascending while **DESC** indicates that the sort is descending.

Considering the following table:

Make	Model	Series	CarID	Meter
Renault	18	RL	4698 SJ 45	123450
Renault	Kangoo	RL	4568 HD 16	56000
Renault	Kangoo	RL	6576 VE 38	12000
Peugeot	106	KID	7845 ZS 83	75600
Peugeot	309	chorus	7647 ABY 82	189500
Ford	Escort	Match	8562 EV 23	

Data Manipulation Language (DML)

• The list of cars sorted by ascending order of the Make is written :

SELECT * FROM CARS

ORDER BY Make ASC

The result is:

Make	Model	Series	CarID	Meter
Ford	Escort	Match	8562 EV 23	
Peugeot	106	KID	7845 ZS 83	75600
Peugeot	309	Chorus	7647 ABY 82	189500
Renault	18	RL	4698 SJ 45	123450
Renault	Kangoo	RL	4568 HD 16	56000

Data Manipulation Language (DML)

• The list of cars sorted by ascending order of the Make and descending order of the Meter is written :

```
SELECT *
FROM CARS
ORDER BY Make ASC,Meter DESC
```

Make	Model	Series	CarID	Meter
Ford	Escort	Match	8562 EV 23	
Peugeot	309	chorus	7647 ABY 82	189500
Peugeot	106	KID	7845 ZS 83	75600
Renault	18	RL	4698 SJ 45	123450
Renault	Kangoo	RL	4568 HD 16	56000
Renault	Kangoo	RL	6576 VE 38	12000

Data Manipulation Language (DML)

- **Grouping**

The SQL clause **GROUP BY** enables to make grouping of results. The major functions associated with grouping are:

AVG: Calculates the average

COUNT: Counts the number of rows

MAX: Calculates the maximum value of a column

MIN: Calculates the minimum value of a column

SUM: Calculates the total of values in a column

Considering our table CARS:

- The average of meters grouped by Make is given by :

```
SELECT Make, AVG (Meter) as Average
 FROM CARS
GROUP BY Make
```

The result is:

Make	Average
Renault	63816.6
Peugeot	132550
Ford	

The clause **HAVING** enables to introduce a logical condition when the clause GROUP BY is used.

• The average of meters grouped by Make and which have an average not null is made by:

```
SELECT Make, AVG (Meter) as Average
FROM CARS
GROUP BY Make
HAVING AVG (Meter) IS NOT NULL
```

The result is:

Make	Average
Renault	63816.6
Peugeot	132550

- **Joint**

The joint enables to join several tables.

Considering the following two tables:

CLIENTS

ClientID	SurName	FirstName	Telephone
001	Uwera	Diane	0788754172
002	Kayiranga	Alain	0788965216
003	Kayitesi	Aimée	0788952141

CARS

CarID	Make	Meter	ClientID
4698 SJ 45	Renault	123450	003
4568 HD 16	Toyota	56000	002
6576 VE 38	Benz	12000	001
7845 ZS 83	Fiat	75600	003
7647 ABY 82	Renault	189500	002
8562 EV 23	Benz		002
8941 UD 61	Fiat		001
7846 AZS 75	Peugeot	21350	003

***The joint is made using the foreign key. If one of the tables doesn't have a primary key, the joint cannot be made

- The list of clients who bought cars is given by:

```
SELECT *
FROM CLIENTS, CARS
WHERE CLIENTS.ClientID =  CARS.ClientID
```

- The list of clients (SurName, FirstName) who bought cars (CarID, Make) manufactured by Benz is written:

```
SELECT SurName,FirstName,CarID,Make
FROM CLIENTS, CARS
WHERE CLIENTS.ClientID= CARS.ClientID
AND Make= "Benz"
```

The result is :

SurName	FirstName	CarID	Make
Uwera	Diane	6576 VE 38	Benz
Kayiranga	Alain	8562 EV 23	Benz

- **Sub Queries**

A query can contain a sub query. The sub query is placed after the clause WHERE or HAVING (If GROUP BY is used).

The syntax is the following:

SELECT ----

FROM ----

WHERE ---- < (SELECT ---- FROM ----)

Or

SELECT ----

FROM ----

HAVING ---- < (SELECT ---- FROM ----)

Considering the following table:

CARS

Make	Model	Series	CarID	Meter
Renault	18	RL	4698 SJ 45	123450
Renault	Kangoo	RL	4568 HD 16	56000
Renault	Kangoo	RL	6576 VE 38	12000
Peugeot	106	KID	7845 ZS 83	75600
Peugeot	309	chorus	7647 ABY 82	189500
Fiat	Punto	GTI	8941 UD 61	80232
Audi	A4	Quattro	7846 AZS 75	21350

The list of cars with a meter less than the average of all the meters is given by :

```
SELECT *
FROM CARS
WHERE Meter < (SELECT AVG (Meter) FROM CARS)
```

The result is :

Make	Model	Series	CarID	Meter
Renault	Kangoo	RL	4568 HD 16	56000
Renault	Kangoo	RL	6576 VE 38	12000
Peugeot	106	KID	7845 ZS 83	75600
Audi	A4	Quattro	7846 AZS 75	21350

Data Manipulation Language (DML)

- The UPDATE Statement

The update statement enables to update data. It also supports the clause **WHERE** like SELECT.

Considering the following table:

CLIENTS

ClientID	Surname	FirstName	Telephone
001	Uwera	Diane	0788754172
002	Kayiranga	Alain	0788965216
003	Kayitesi	Aimée	07889454850

If the telephone number of Kayitesi has changed,we will update it using UPDATE as follows:

UPDATE CLIENTS

SETTelephone= "0788952141"

WHERE ClientID="003"

The result will be:

ClientID	Surname	FirstName	Telephone
001	Uwera	Diane	0788754172
002	Kayiranga	Alain	0788965216
003	Kayitesi	Aimée	0788952141

• If you want to add 5% on the initial prices of the products which cost more than 1000;the code is:

UPDATE PRODUCTS

SET Price = Price + Price*0.05

WHERE Price > 1000

*** The statement UPDATE also supports sub queries.The Syntax is the following:

UPDATETable x

SET……..

WHERE ……..(SELECT……FROM……)

Data Manipulation Language (DML)

- **The INSERT Statement**

The insert statement enables to add rows on tables.

The Syntax is the following:

INSERT INTOTablexVALUES ('Val1','Val2',……,'Valn')

Or

INSERT INTOTablex (Column1,Column2,….,Column n)VALUES

(Val1','Val2',……,'Valn') {*When you need specific columns*}

Considering the table following table:

PRODUCTS

ProductID	Designation	Price
ART01	Computer	450000
ART02	Printer	150000

Data Manipulation Language (DML)

INSERT INTO PRODUCTSVALUES ('ART03','Scanner','250000')

The result is:

ProductID	Designation	Price
ART01	Computer	450000
ART02	Printer	150000
ART03	Scanner	250000

Data Manipulation Language (DML)

The result is:

ProductID	Designation	Price
ART02	Printer	150000
ART03	Scanner	250000

*** The statement DELETE also supports sub queries. The Syntax is the following:

DELETE FROM Table x

WHERE ……..(SELECT……FROM……)

Dat Maanipulation Language (DML)

- **The DELETE Statement**

The command delete enables to delete data in a table. It also supports the clause **WHERE** like SELECT and UPDATE.

• The deletion of all the data in the table PRODUCTS is written:

DELETE FROM PRODUCTS

Considering the following table:

ProductID	Designation	Price
ART01	Computer	450000
ART02	Printer	150000
ART03	Scanner	250000

• The deletion of the product Computer identified by ART01 is written as follows :

DELETE FROM PRODUCTS

WHERE ProductID='ART01'

Data Control Language (DCL)

- A **data control language** (DCL) is a syntax similar to a computer programming **language** used to **control** access to **data** stored in a database (Authorization).
- **The GRANT command** is used by administrators to add new permissions to a database user.

```
GRANT [privilege]
ON [object]
TO [user]
[WITH GRANT OPTION]
```

.**User** may be any database user.

.**Privilege** may be either the keyword **ALL** (to grant a wide variety of permissions) or a specific database permission or set of permissions.
.Examples include CREATE DATABASE, SELECT, INSERT, UPDATE, DELETE, EXECUTE,and CREATEVIEW.

Example:

```
GRANT SELECT
ON HR.employees
TO Joe
```

.**Object** may be any database object.
.Typically the object will be either a database, function, stored procedure, table or view.

Data Control Language (DCL)

- The **DENY command** may be used to explicitly prevent a user from receiving a particular permission.
  - This is helpful when a user may be a member of a role or group that is granted a permission and you want to prevent that user from inheriting the permission by creating an exception.
- The syntax for this command is as follows:

```
DENY [permission]
ON [object]
TO [user]
```

.The parameters for the DENY command are identical to those used for the GRANT command.

Example:

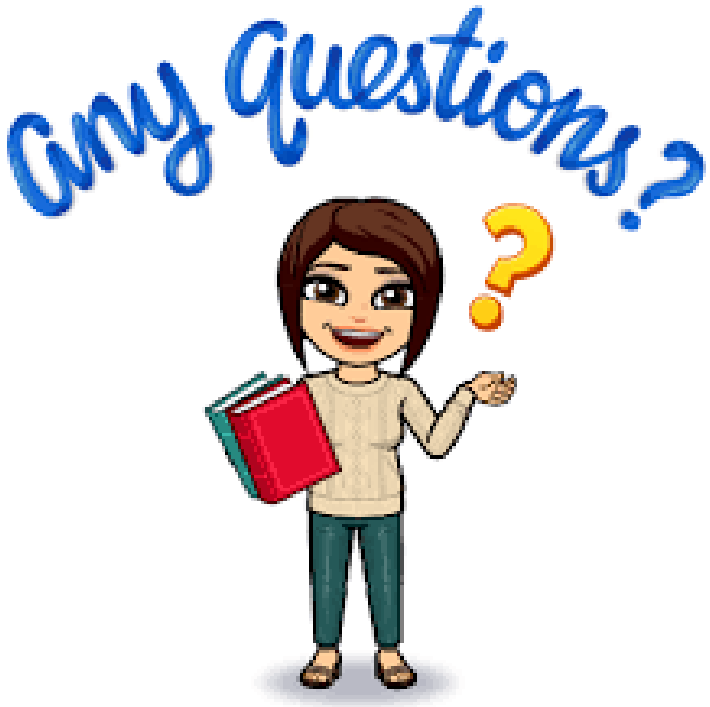
```
DENY DELETE
ON HR.employees
TO Matthew
```

.For example, if you wished to ensure that **Matthew** would never receive the ability to delete information from the employees table, you would issue the following command:

Transaction Control (TCL)

- Statements are used to manage the changes made by DML statements.
- It allows statements to be grouped together into logical transactions.
  - Some examples: **COMMIT** - COMMIT saves the transaction on the database. The transaction can be insert, delete or update. Once the COMMIT is issued, the changes are saved permanently in the database. It cannot be undone.
- **SAVEPOINT** - savepoint command is used to temporarily save a transaction so that you can rollback to that point whenever necessary.


THANK
YOU


Any questions?