Bases de données

Prof. Giovanna Di Marzo Serugendo
Assistant: Mohammad Parhizkar
mohammad.parhizkar@unige.ch

Semestre: printemps 2019

la session 1- 25/02/2019

Travaux pratique

- Objectif: consolider SQL + pratiquer PL/SQL, MySQL, PHP
- Correction: au cours du séminaire
- Obligatoire:

Compte pour 1/3 de la note finale du cours (l'examen les autres 2/3)!

Travaux Pratique

- **Rendu:**
  - Merci de le soumettre sur la **plateforme:**
    - Dans un seul fichier (1 par TP)
    - Nom & prénom
    - Code pl/sql + résultat affiché (screenshot)

- Questions et commentaires: par email
  - mohammad.parhizkar@unige.ch
  - dans le sujet: [bdd/19/tp#]

UNIVERSITÉ DE GENÈVE
CENTRE UNIVERSITAIRE D'INFORMATIQUE

Liens Utiles…

SQL	Oracle Database SQL Language Reference 11g Release 2 http://docs.oracle.com/cd/E14072_01/server.112/e10592.pdf
	Forum SQL & PL/SQL https://forums.oracle.com/forums/forum.jspa?forumID=75
PL/SQL	PL/SQL User's Guide and Reference http://docs.oracle.com/cd/E11882_01/appdev.112/e25519.pdf
	Forum SQL & PL/SQL https://forums.oracle.com/forums/forum.jspa?forumID=75
Oracle Express Edition (XE)	[serveur Oracle léger] http://www.oracle.com/technetwork/database/express-edition/overview/
Autres outils	SQL Developer [client Oracle: dév. BD; édition PL/SQL] http://www.oracle.com/technology/software/products/sql/index.html
	Crimson Editor [éditeur de texte; reconnaissance PL/SQL] http://www.crimsoneditor.com/
Test Oracle XE	Browser based development [démo] http://apex.oracle.com

Oracle SQL Developer

Tutoriel d’installation

SQL Developer permet de créer et d'exécuter des requêtes et des scripts SQL, de déboguer des exécutions de script, d'afficher des résultats et de gérer des bases de données.

Comment installer l'application

Oracle SQL Developer

Pour fonctionner, SQL Developer utilise un **JDK**. Si vous n'avez pas déjà une version du JDK supérieure ou égale à la 1.6.11, ou si vous avez la version 1.7.n, *téléchargez la version contenant le JDK.*

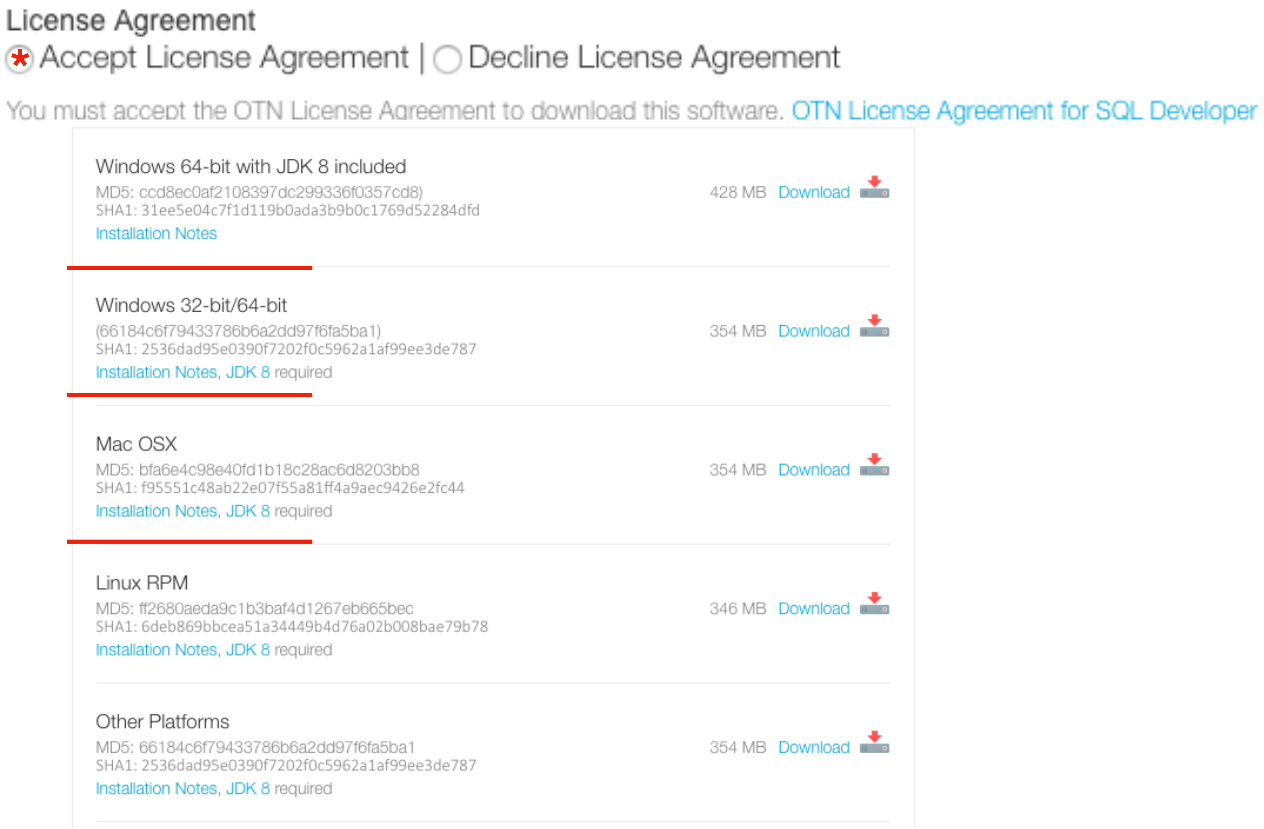
Comment installer l'application

Oracle SQL Developer

les étapes:

1. https://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html ou depuis l'accueil du site d'Oracle.

2.



License Agreement

● Accept License Agreement | ○ Decline License Agreement

You must accept the OTN License Agreement to download this software. OTN License Agreement for SQL Developer

3.

Windows 64-bit with JDK 8 included
MD5: ccd8ec0af2108397dc299336f0357cd8)
SHA1: 31ee5e04c7f1d119b0ada3b9b0c1769d52284dfd
Installation Notes
428 MB Download

Windows 32-bit/64-bit
(66184c6f79433786b6a2dd97f6fa5ba1)
SHA1: 2536dad95e0390f7202f0c5962a1af99ee3de787
Installation Notes, JDK 8 required
354 MB Download

Mac OSX
MD5: bfa6e4c98e40fd1b18c28ac6d8203bb8
SHA1: f95551c48ab22e07f55a81ff4a9aec9426e2fc44
Installation Notes, JDK 8 required
354 MB Download

Linux RPM
MD5: ff2680aeda9c1b3baf4d1267eb665bec
SHA1: 6deb869bbcea51a34449b4d76a02b008bae79b78
Installation Notes, JDK 8 required
346 MB Download

Other Platforms
MD5: 66184c6f79433786b6a2dd97f6fa5ba1
SHA1: 2536dad95e0390f7202f0c5962a1af99ee3de787
Installation Notes, JDK 8 required
354 MB Download

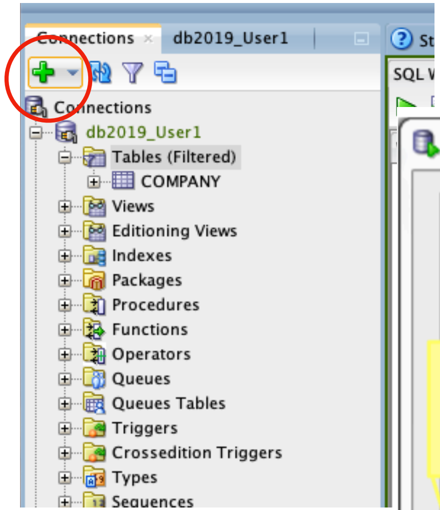
Oracle SQL Developer

- Lancez SQL Developer

Créer une nouvelle connexion:

Faites un clic droit sur l'onglet « Connexion » puis cliquez sur *Nouvelle connexion:*

étap 1.

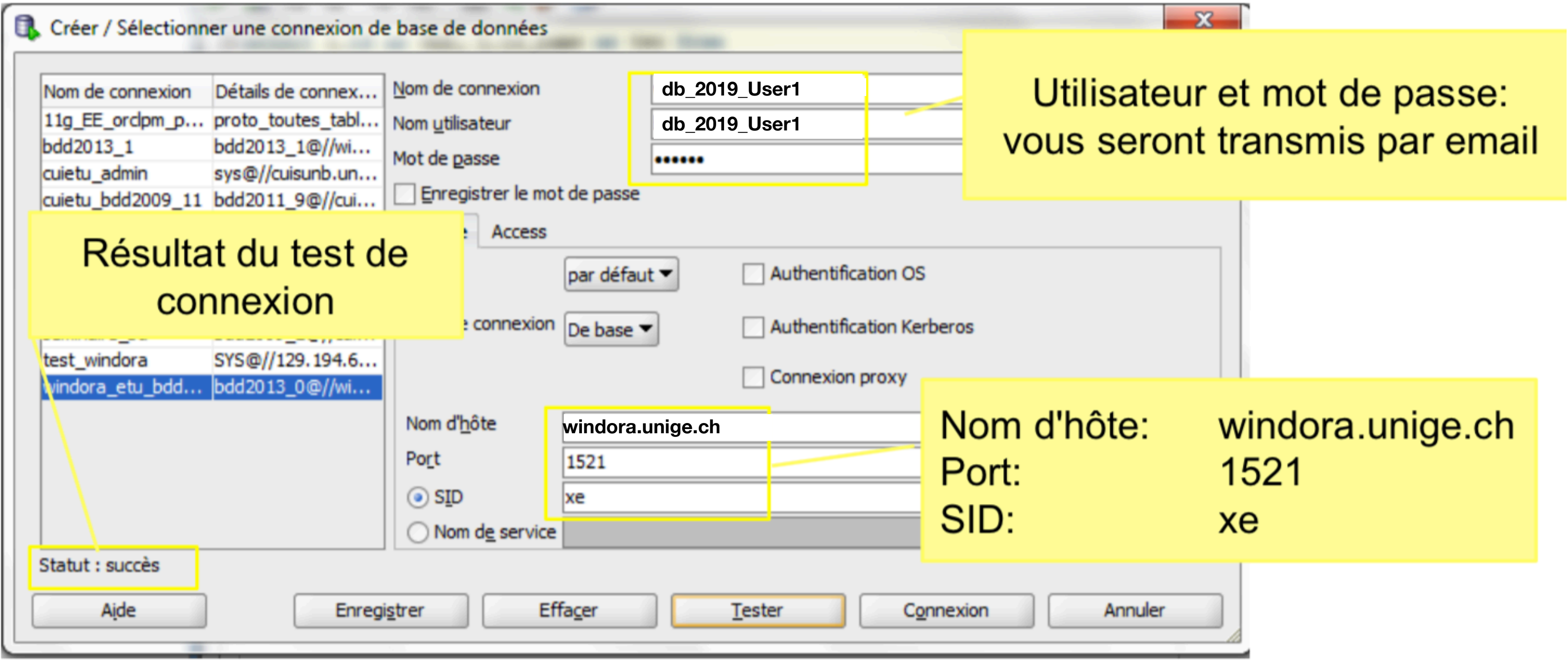


Connections × | db2019_User1 | St

SQL W

- Connections
  - db2019_User1
    - Tables (Filtered)
      - COMPANY
    - Views
    - Editioning Views
    - Indexes
    - Packages
    - Procedures
    - Functions
    - Operators
    - Queues
    - Queues Tables
    - Triggers
    - Crossedition Triggers
    - Types
    - Sequences

étap 2.



Créer / Sélectionner une connexion de base de données

Nom de connexion	Détails de connex...
11g_EE_orclpm_p...	proto_toutes_tabl...
bdd2013_1	bdd2013_1@//wi...
cuietu_admin	sys@//cuisunb.un...
cuietu_bdd2009_11	bdd2011_9@//cui...
test_windora	SYS@//129.194.6...
windora_etu_bdd...	bdd2013_0@//wi...

Nom de connexion: db_2019_User1

Nom utilisateur: db_2019_User1

Mot de passe: ••••••

☐ Enregistrer le mot de passe

Access

par défaut ▾ ☐ Authentification OS

connexion: De base ▾ ☐ Authentification Kerberos

☐ Connexion proxy

Nom d'hôte: windora.unige.ch

Port: 1521

● SID: xe

○ Nom de service

Statut : succès

Utilisateur et mot de passe: vous seront transmis par email

Résultat du test de connexion

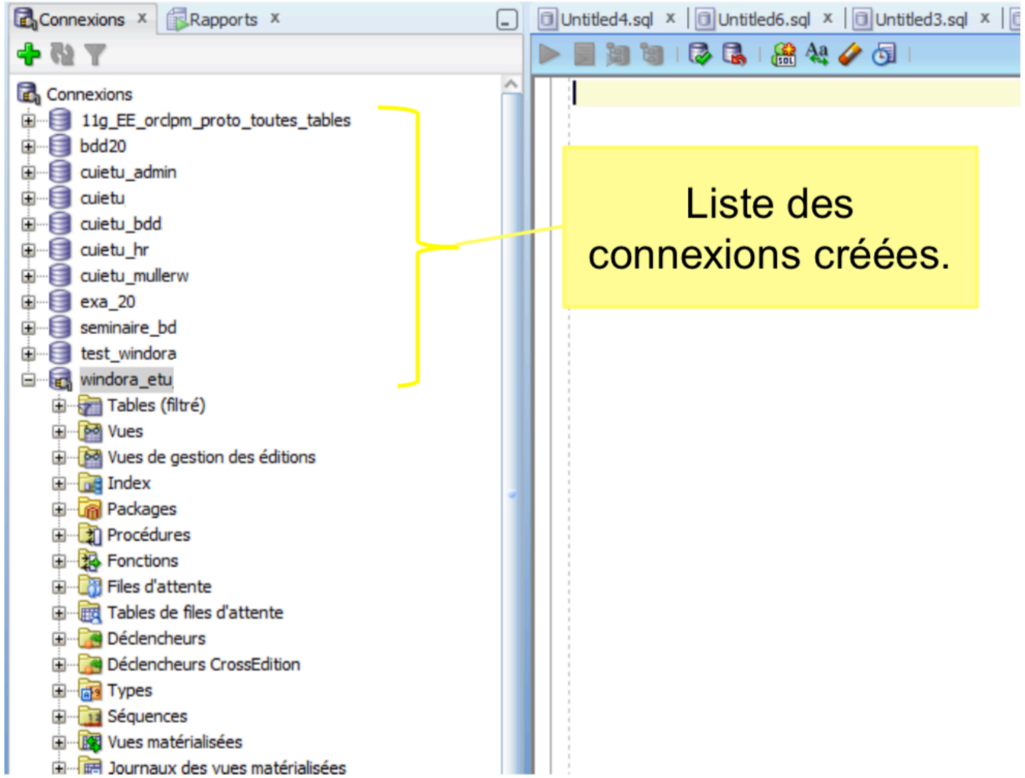
Nom d'hôte:	windora.unige.ch
Port:	1521
SID:	xe

étap 3. Cliquez sur *Tester* puis sur Enregistrer.

Oracle SQL Developer

Se connecter:

- Double-cliquer sur le nom de la connexion créée.

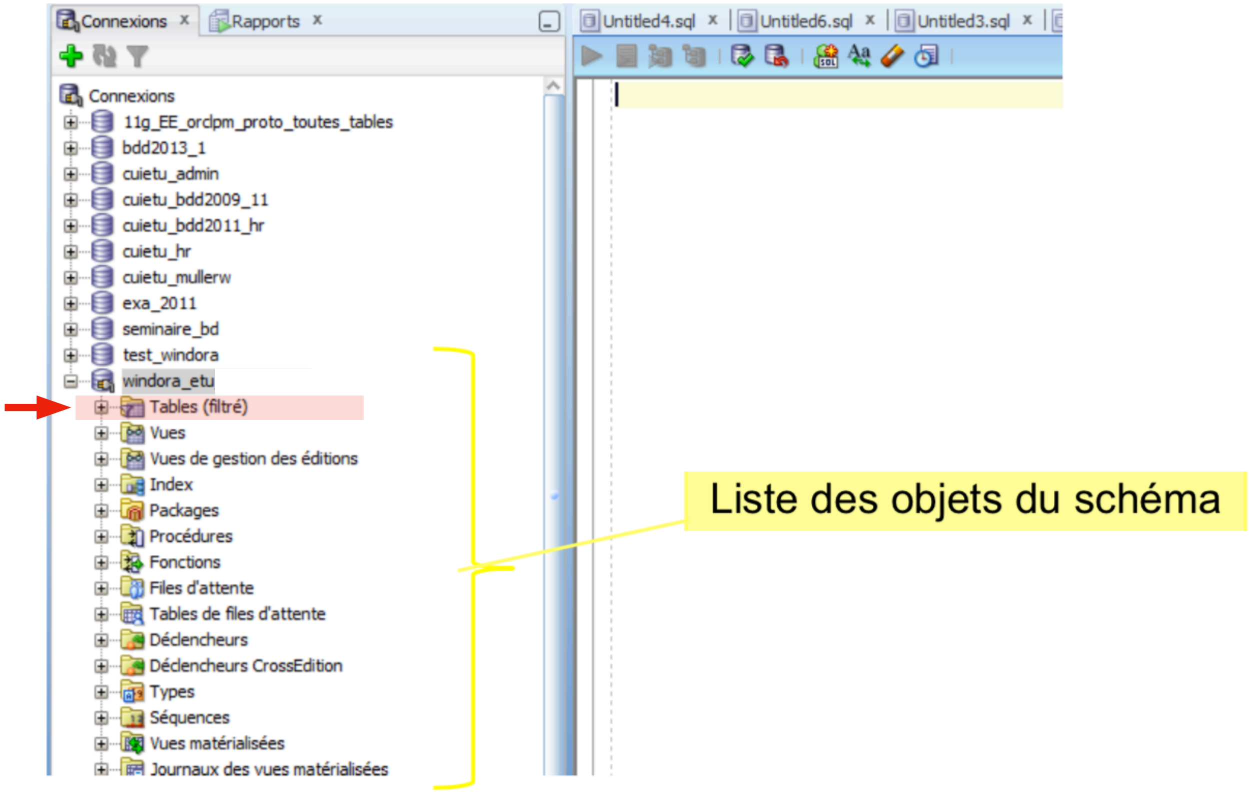


Connexions
- 11g_EE_orclpm_proto_toutes_tables
- bdd20
- cuietu_admin
- cuietu
- cuietu_bdd
- cuietu_hr
- cuietu_mullerw
- exa_20
- seminaire_bd
- test_windora
- windora_etu
  - Tables (filtré)
  - Vues
  - Vues de gestion des éditions
  - Index
  - Packages
  - Procédures
  - Fonctions
  - Files d'attente
  - Tables de files d'attente
  - Déclencheurs
  - Déclencheurs CrossEdition
  - Types
  - Séquences
  - Vues matérialisées
  - Journaux des vues matérialisées

Liste des connexions créées.

Oracle SQL Developer

Naviguer dans les objets du schéma:



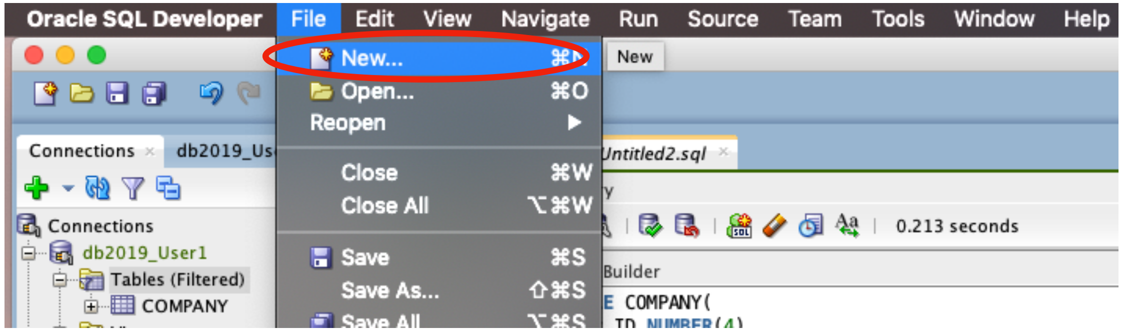
Connexions
- 11g_EE_orclpm_proto_toutes_tables
- bdd2013_1
- cuietu_admin
- cuietu_bdd2009_11
- cuietu_bdd2011_hr
- cuietu_hr
- cuietu_mullerw
- exa_2011
- seminaire_bd
- test_windora
- windora_etu
  - Tables (filtré)
  - Vues
  - Vues de gestion des éditions
  - Index
  - Packages
  - Procédures
  - Fonctions
  - Files d'attente
  - Tables de files d'attente
  - Déclencheurs
  - Déclencheurs CrossEdition
  - Types
  - Séquences
  - Vues matérialisées
  - Journaux des vues matérialisées

Liste des objets du schéma

Comment créer une nouvelle page?

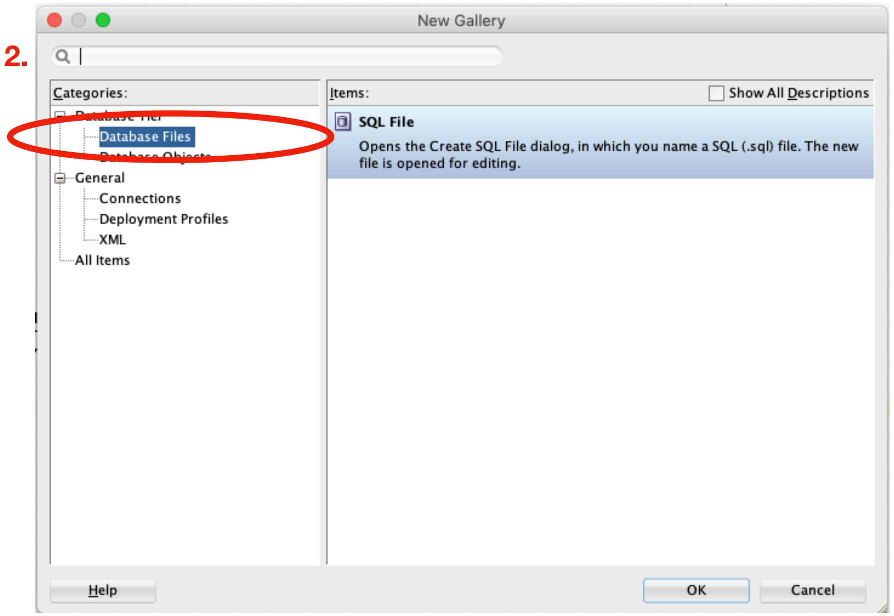
un SQL fichier

étap 1.



```
E COMPANY(
  ID NUMBER(4)
```

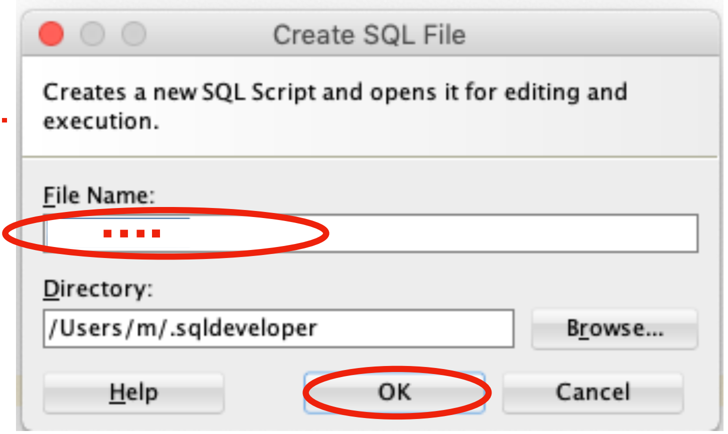
étap 2.



SQL File

Opens the Create SQL File dialog, in which you name a SQL (.sql) file. The new file is opened for editing.

étap 3.



Creates a new SQL Script and opens it for editing and execution.

File Name: ....

Directory: /Users/m/.sqldeveloper

UNIVERSITÉ DE GENÈVE

CENTRE UNIVERSITAIRE D'INFORMATIQUE

SQL?

What is SQL?

(pronounced "ess-que-el")

- Stands for Structured Query Language.
- SQL is used to communicate with a database.
- We use it to perform tasks such as update data on a database, or retrieve data from a database.
- Some common relational database management systems that use SQL are: **Oracle**, Sybase, Microsoft SQL Server, Access, Ingres, etc.

Oracle is the most widely used Relationship Database Management System (RDBMS) in the word.

Golden Rules that Define a Relational Database

- All information is represented as value in a table.
- All data items are accessible by giving the table name, column and row.
- NULL values are valid datatype, for missing information.
- SQL is primary interface language supported by the relational model.
- Logical aspect of the database is totally separated from physical aspects of it.
- …

Essential Oracle 7: Tom Luers

Base table vs. User view

- Base table:

- The tables are the foundation which stores information about database.
- Tables are accessible directly by oracle itself.
- The data stored in these tables is stored in Oracle's own format.

- User accessible views:

- The views summarize and present the information.
- The views decode the cryptic base table information for the users.
- Most users have access to the views rather than having access to the base tables

Datatypes

They are predefined statistic set of information

- **Char:** the length of the string (alphanumeric data) ranges from 1 to 255. It has fixed length character string. If the inserted value is shorter than the fixed- length variable, Oracle will put blank space. If it is longer, Oracle will returns an error notification.

Examples:

User_Name char(15)

Data example: TM33P93ZRT

- **varchar2:** It is a variable-length character string. If the inserted value is shorter in length, Oracle will **not** add blank space. However, if the inserted value is longer, the user will get an error.

Examples:

User_Name varchar2(15)

Data example: TM33P93ZRT

- **varchar:** Currently, they are the same. However, previously they were different in terms of supporting distinguish between NULL and empty string.

Datatypes

- **number / int:** To store fixed and floating point numbers, so it can be defined with a precision and a scale. The numbers can be stored with up to 38 digits of precision and a scale ranging from -84 to 127.

Examples:

Product_price number (9)

Data example: 998765900

also…

Product_price number(7,3)

Data example: 568332.004

(6 is precision and 2 is scale)

- **date:** To store date and time which covers year, month, day, the hour, the minutes and the seconds. The format can be different in different versions of Oracle.

Examples:

- Format for date: DD-MON-YY
- Format for time: HH:MM:SS

Datatypes

These two datatypes are intended for binary data:

- **raw**
- **long raw**

- **rowid:** It corresponds to the physical address of a row in Oracle table.
  - Rowid is fastest way to access to a row in a table.
  - We use it to see how tables are organized.
  - They are unique identifiers for the rows in the tables.

- **rownum**
- **long**

Datatypes Conversion

— Oracle does it automatically *under certain circumstances*.

— However, we can use these functions:

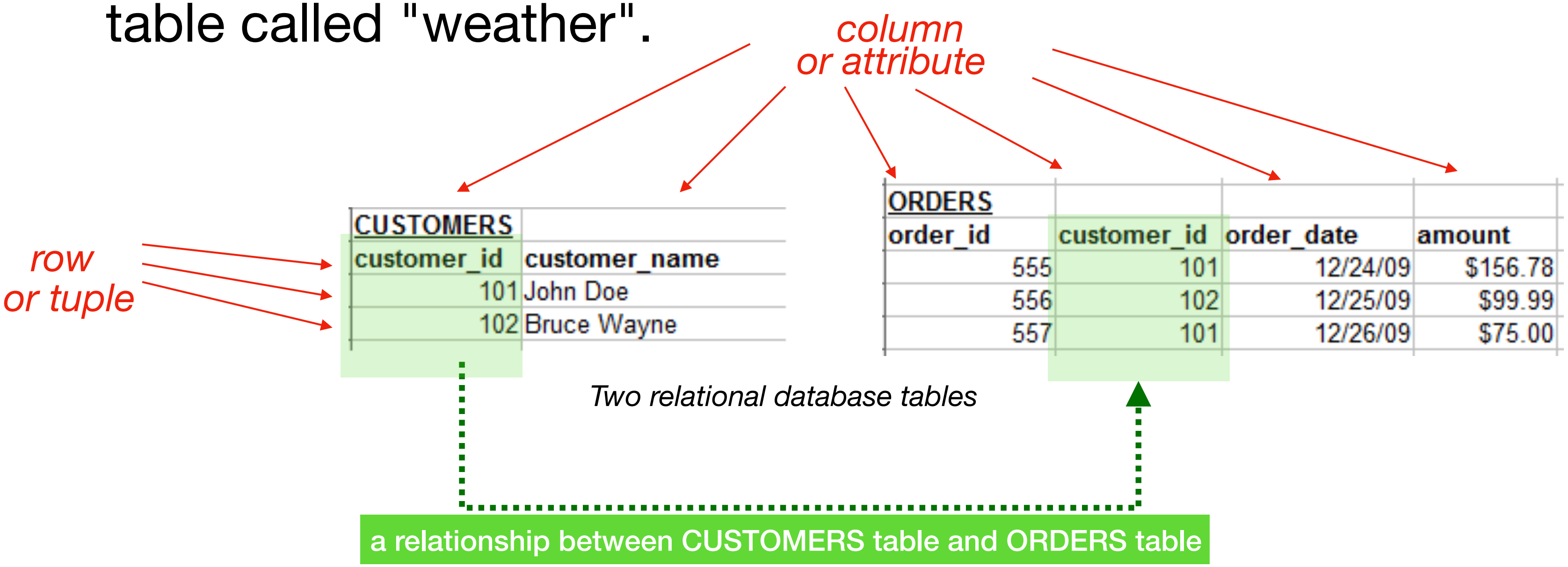
- to_char()
- to_date()
- to_number()
- to_label()
- chartorowid()
- rowidtochar()

— We can also define and use our own functions…

- varchar2 or char to number
- number to varchar2
- varchar2 or char to date

Tables Basics

- A relational database system contains one or more objects called tables.
- The information are stored in the tables.
- Tables are uniquely identified by their names.Tables comprised of columns and rows:
  - Columns contain the column name, data type, and any other attributes for the column.
  - Rows contain the records or data for the columns. Here is a sample table called "weather".



CUSTOMERS

customer_id	customer_name
101	John Doe
102	Bruce Wayne

ORDERS

order_id	customer_id	order_date	amount
555	101	12/24/09	$156.78
556	102	12/25/09	$99.99
557	101	12/26/09	$75.00

Two relational database tables

a relationship between CUSTOMERS table and ORDERS table

Creating Tables

PL/SQL syntax:

```
CREATE TABLE table_name
(
column1 data_type(size) constraint,
column2 data_type(size) constraint,
column3 data_type(size) constraint,
....
);
```

table_name: name of the table.
column1 name of the first column.
data_type: Type of data we want to store in the particular column.
For example, **int** for integer data.

size: Size of the data we can store in a particular column. For example if for a column we specify the data_type as int and size as 10 then this column can store an integer number of maximum 10 digits.

Constraint: A rule that restricts the data value for one or more columns in a table.

Creating Tables

Example 1:

```
CREATE TABLE Customers
(
CustomerId NUMBER (3) NOT NULL,
CustomerName VARCHAR2(20) NOT NULL,
Canton VARCHAR2(20) NOT NULL,
Sex NUMBER(1),
Age NUMBER (3),
Status CHAR(1)
);
```

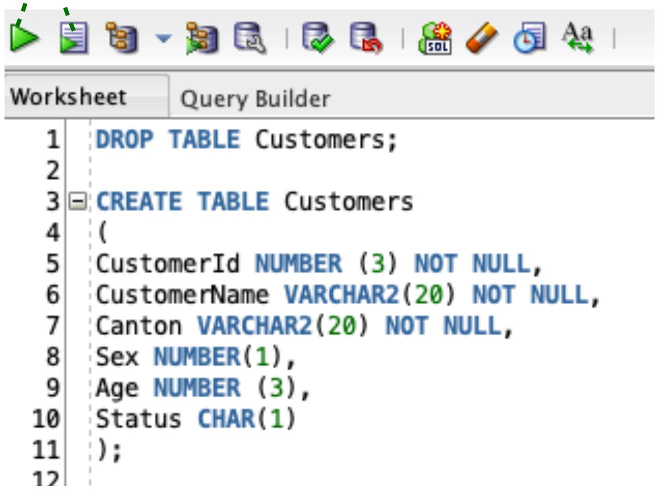
id	Name	Canton	Sex	Age	Status

```
# femal= 1, male=2, other=3
# marital status: Single=S, Married=M, Divorsed=D
```

Execute your code:

to run the selected segment

to run the whole code together

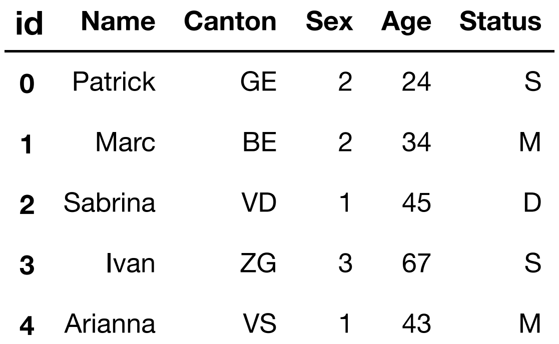


```
DROP TABLE Customers;

CREATE TABLE Customers
(
CustomerId NUMBER (3) NOT NULL,
CustomerName VARCHAR2(20) NOT NULL,
Canton VARCHAR2(20) NOT NULL,
Sex NUMBER(1),
Age NUMBER (3),
Status CHAR(1)
);
```

INSERT Statement

Description:

The PL/SQL INSERT statement is used to place a record or multiple records into a specific table in Oracle.



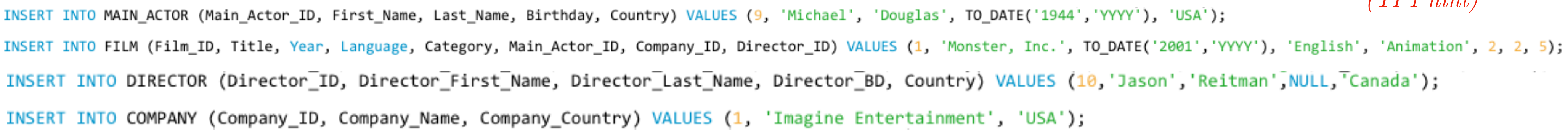
id	Name	Canton	Sex	Age	Status
0	Patrick	GE	2	24	S
1	Marc	BE	2	34	M
2	Sabrina	VD	1	45	D
3	Ivan	ZG	3	67	S
4	Arianna	VS	1	43	M

Syntax:

The syntax for the Oracle INSERT statement when inserting a single record using the VALUES keyword is:

```
INSERT INTO table
(column1, column2, ... column_n )
VALUES
(expression1, expression2, ... expression_n );
```

(TP1 hint)

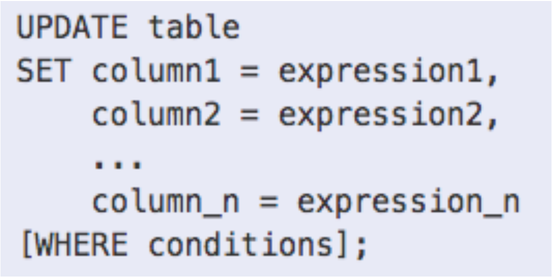


```
INSERT INTO MAIN_ACTOR (Main_Actor_ID, First_Name, Last_Name, Birthday, Country) VALUES (9, 'Michael', 'Douglas', TO_DATE('1944','YYYY'), 'USA');
INSERT INTO FILM (Film_ID, Title, Year, Language, Category, Main_Actor_ID, Company_ID, Director_ID) VALUES (1, 'Monster, Inc.', TO_DATE('2001','YYYY'), 'English', 'Animation', 2, 2, 5);
INSERT INTO DIRECTOR (Director_ID, Director_First_Name, Director_Last_Name, Director_BD, Country) VALUES (10,'Jason','Reitman',NULL,'Canada');
INSERT INTO COMPANY (Company_ID, Company_Name, Company_Country) VALUES (1, 'Imagine Entertainment', 'USA');
```

UPDATE Statement

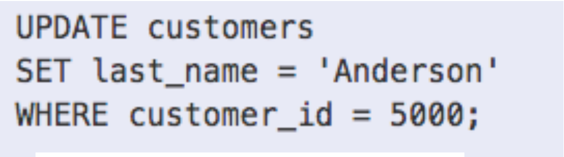
to update existing records in a table

Syntax:



```
UPDATE table
SET column1 = expression1,
    column2 = expression2,
    ...
    column_n = expression_n
[WHERE conditions];
```

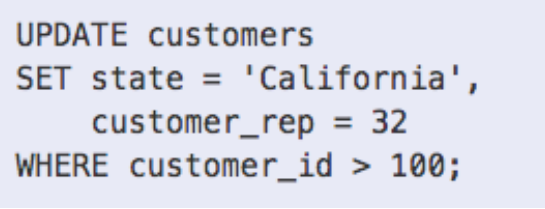
Example 1:



```
UPDATE customers
SET last_name = 'Anderson'
WHERE customer_id = 5000;
```

Update one column, one row

Example 2:



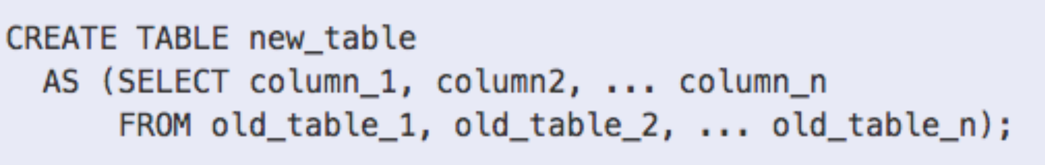
```
UPDATE customers
SET state = 'California',
    customer_rep = 32
WHERE customer_id > 100;
```

Update multiple columns, multiple rows

CREATE TABLE AS …

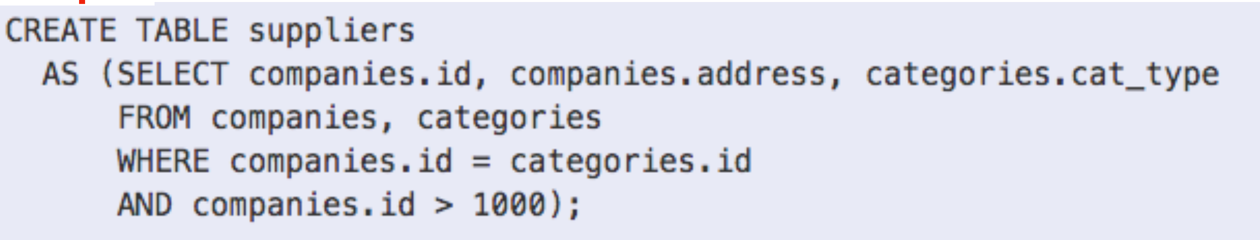
Create a table by copying selected columns from multiple tables

Syntax:



```
CREATE TABLE new_table
  AS (SELECT column_1, column2, ... column_n
      FROM old_table_1, old_table_2, ... old_table_n);
```

Example:



```
CREATE TABLE suppliers
  AS (SELECT companies.id, companies.address, categories.cat_type
      FROM companies, categories
      WHERE companies.id = categories.id
      AND companies.id > 1000);
```

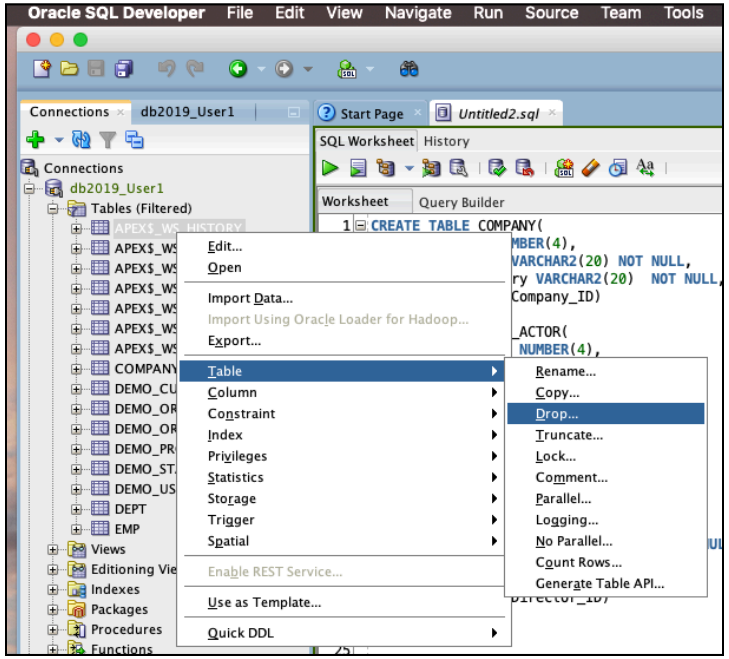
Remove a table

Two approaches:

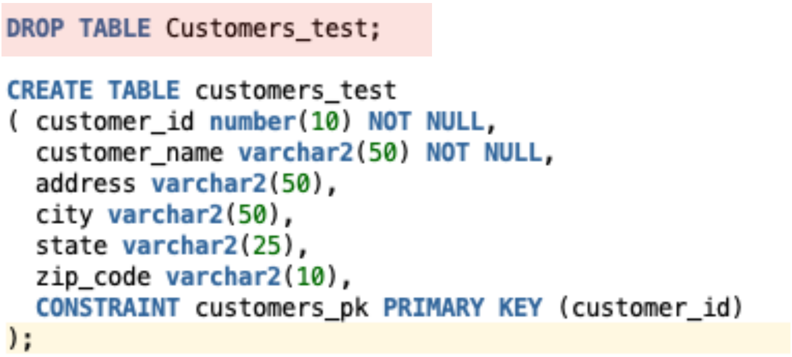
1 We can use the ‘DROP TABLE ….’ statement to remove a table and all its data from the database.

```
DROP TABLE customers;
```

2



Example:



```
DROP TABLE Customers_test;

CREATE TABLE customers_test
( customer_id number(10) NOT NULL,
  customer_name varchar2(50) NOT NULL,
  address varchar2(50),
  city varchar2(50),
  state varchar2(25),
  zip_code varchar2(10),
  CONSTRAINT customers_pk PRIMARY KEY (customer_id)
);
```

Output:

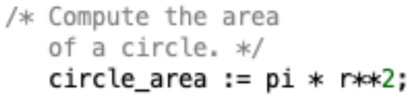
```
Table CUSTOMERS_TEST dropped.


Table CUSTOMERS_TEST created.
```

Comments in PL/SQL Code

Comments in your program describe the purpose and use of code segments and help us in terms of the readability.

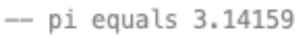
Two approaches:

1



```
/* Compute the area
   of a circle. */
   circle_area := pi * r**2;
```

2

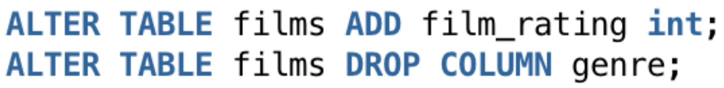


```
-- pi equals 3.14159
```

ALTER TABLE Statement

1. To add, delete, or modify (e.g. datatype) one or more columns in an existing table

2. To add or drop various constraints on an existing table



```
ALTER TABLE films ADD film_rating int;
ALTER TABLE films DROP COLUMN genre;
```

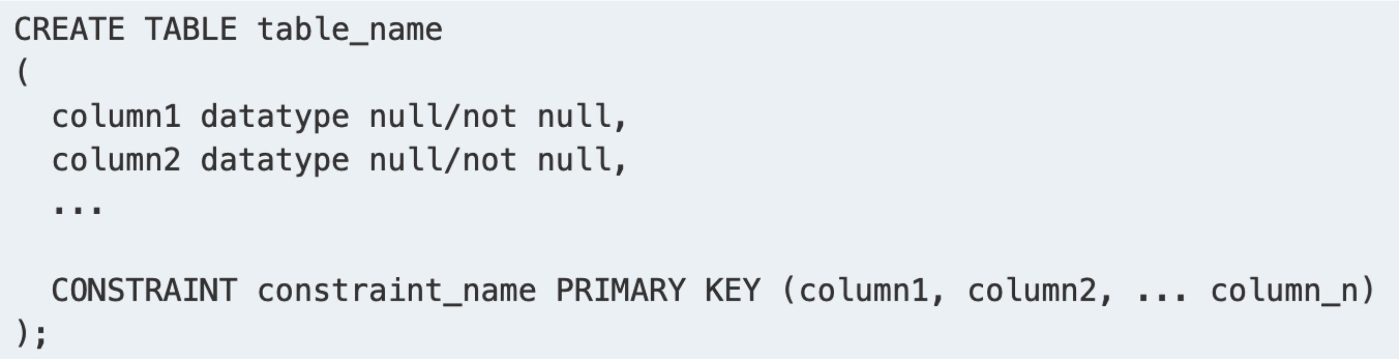
Primary Key

A **primary key** is:

- A single field or combination of fields that uniquely defines a record.
- The fields that are part of the primary key can not contain a null value.
- A table can have only one primary key.

- In Oracle, a primary key can not contain more than 32 columns.
- A primary key can be defined in either a CREATE TABLE statement or an ALTER TABLE statement.

Syntax:



```
CREATE TABLE table_name
(
  column1 datatype null/not null,
  column2 datatype null/not null,
  ...

  CONSTRAINT constraint_name PRIMARY KEY (column1, column2, ... column_n)
);
```

Primary Key

In this example, a primary key on the customer table called customer_pk has been created. It consists of only one field: the customer_id field.

Example 1:

```
CREATE TABLE customer
(
  customer_id numeric(10) not null,
  customer_name varchar2(50) not null,
  customer_address varchar2(50),
  CONSTRAINT customer_pk PRIMARY KEY (customer_id)
);
```

We could also create a primary key with more than one field as in the example below:

Example 2:

```
CREATE TABLE customer
(
  customer_id numeric(10) not null,
  customer_name varchar2(50) not null,
  customer_address varchar2(50),
  CONSTRAINT customer_pk PRIMARY KEY (customer_id, customer_name)
);
```

Primary Key

Create Primary Key by using ALTER TABLE statement

Syntax:

```
ALTER TABLE table_name
ADD CONSTRAINT constraint_name PRIMARY KEY (column1, column2, ... column_n);
```

In this example, we've created a primary key on the existing supplier table called customer_pk. It consists of the field called customer_id.

Example 1:

```
ALTER TABLE customer
ADD CONSTRAINT customer_pk PRIMARY KEY (customer_id);
```

We could also create a primary key with more than one field as in the example below:

Example 2:

```
ALTER TABLE customer
ADD CONSTRAINT customer_pk PRIMARY KEY (customer_id, customer_name);
```

Primary Key

Drop Primary Key

You can always drop a primary key in Oracle using the ALTER TABLE statement.

Syntax:

```
ALTER TABLE table_name
DROP CONSTRAINT constraint_name;
```

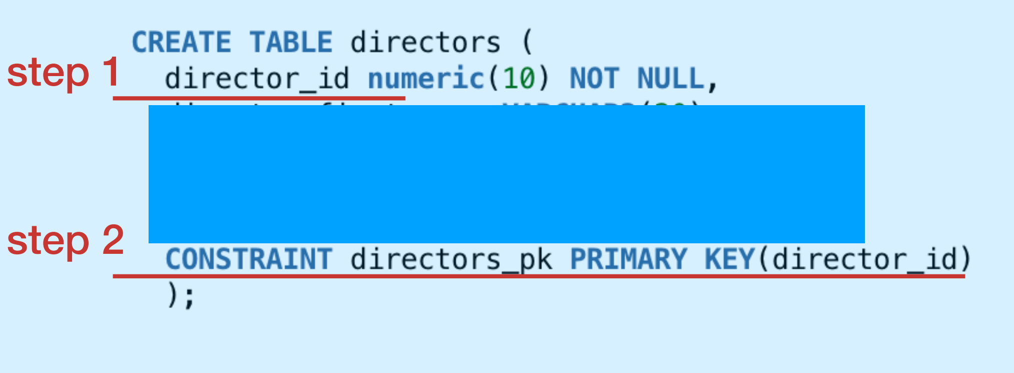
Example:

```
ALTER TABLE customer
DROP CONSTRAINT customer_pk;
```

Foreign key definition

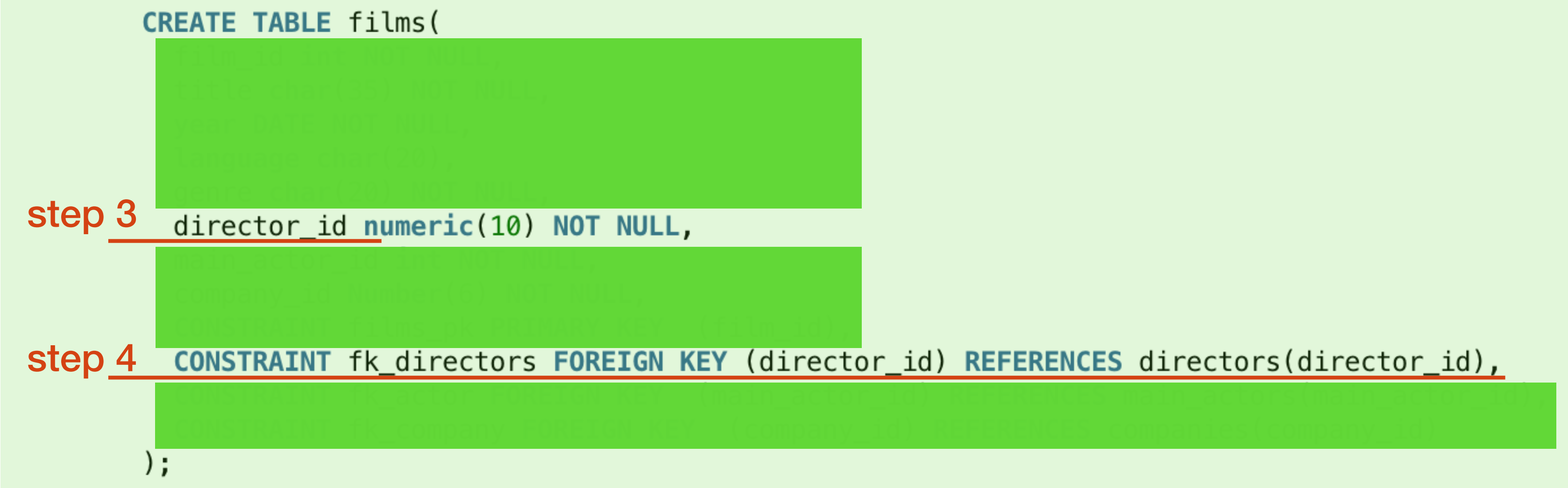
- A foreign key is a way to make integrity and within your Oracle database.
- A foreign key means that values in one table must also appear in another table.
- The referenced table is called the *parent table* while the table with the foreign key is called the *child table*.
- The foreign key in the child table will generally reference a primary key in the parent table.
- A foreign key can be defined in either a *CREATE TABLE* statement or an *ALTER TABLE* statement.

Foreign key: PL/SQL steps (TP1 hint)



```
         CREATE TABLE directors (
step 1     director_id numeric(10) NOT NULL,
[redacted]
step 2     CONSTRAINT directors_pk PRIMARY KEY(director_id)
           );
```

- In this example, we've created a primary key on the directors table called *directors_pk*.
- Then we've created a foreign key called *fk_directos* on the films table that references the directors table based on the director_id field.



```
         CREATE TABLE films(
[redacted]
step 3     director_id numeric(10) NOT NULL,
[redacted]
step 4     CONSTRAINT fk_directors FOREIGN KEY (director_id) REFERENCES directors(director_id),
[redacted]
         );
```

FK: Using an ALTER TABLE statement

Syntax

The syntax for creating a foreign key in an ALTER TABLE statement is:

```
ALTER TABLE table_name
ADD CONSTRAINT constraint_name
   FOREIGN KEY (column1, column2, ... column_n)
   REFERENCES parent_table (column1, column2, ... column_n);
```

Example:

```
ALTER TABLE films
ADD CONSTRAINT fk_directors
  FOREIGN KEY (directo_id)
  REFERENCES direcotrs(director_id);
```

- The most commonly used statement in SQL
- To fetch data from a database.

Select

If we want to retrieve attributes **id** and **Name** of all customers, the query will be:

id	Name	Canton	Sex	Age	Status
0	Patrick	GE	2	24	S
1	Marc	BE	2	34	M
2	Sabrina	VD	1	45	D
3	Ivan	ZG	3	67	S
4	Arianna	VS	1	43	M

Example 1:

```
SELECT id, NAME FROM Customers;
```

Output 1:

id	Name
0	Patrick
1	Marc
2	Sabrina
3	Ivan
4	Arianna

Select

If we want to retrieve attributes **id** and **Name** of all customers whose id is greater than 2, the query will be:

Example 2:

```
SELECT id, NAME FROM Customers WHERE id>2;
```

Output 2:

id	Name
2	Sabrina
3	Ivan
4	Arianna

Example 3:

```
SELECT ∗ FROM Customers WHERE id>2;
```

Output 3:

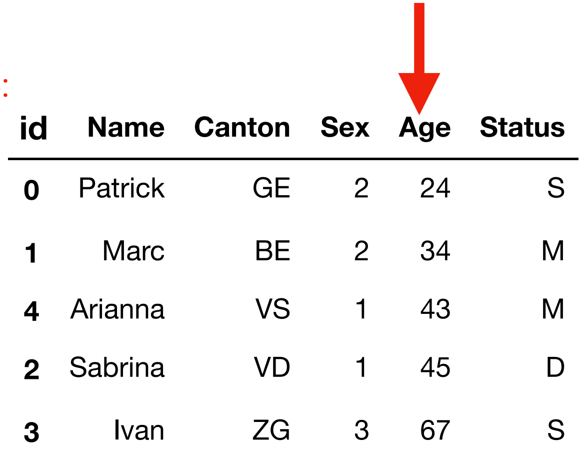
id	Name	Canton	Sex	Age	Status
2	Sabrina	VD	1	45	D
3	Ivan	ZG	3	67	S
4	Arianna	VS	1	43	M

Select: ORDER BY, DISTINCT

Example 5:

```
SELECT * FROM Customers ORDER BY Age;
```

Output 5:

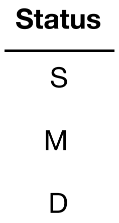


id	Name	Canton	Sex	Age	Status
0	Patrick	GE	2	24	S
1	Marc	BE	2	34	M
4	Arianna	VS	1	43	M
2	Sabrina	VD	1	45	D
3	Ivan	ZG	3	67	S

Example 6:

```
SELECT DISTINCT Status FROM Customers;
```

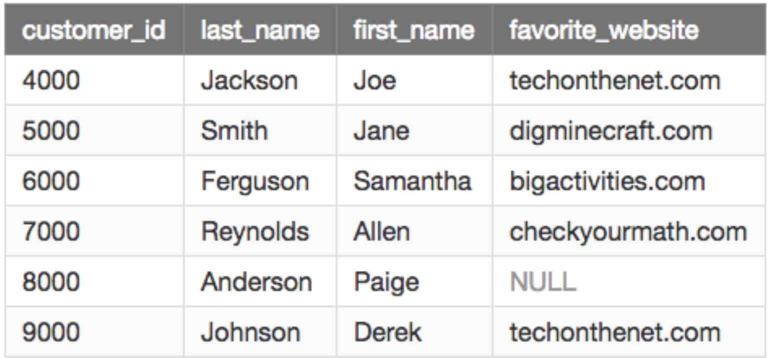
Output 6:



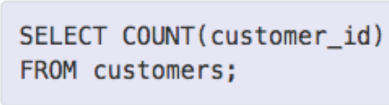
Status
S
M
D

Count Function

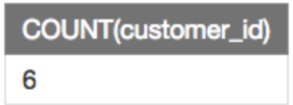
Example 1:



customer_id	last_name	first_name	favorite_website
4000	Jackson	Joe	techonthenet.com
5000	Smith	Jane	digminecraft.com
6000	Ferguson	Samantha	bigactivities.com
7000	Reynolds	Allen	checkyourmath.com
8000	Anderson	Paige	NULL
9000	Johnson	Derek	techonthenet.com

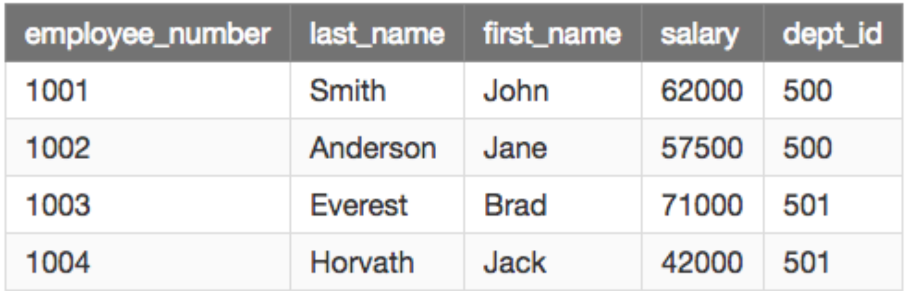


```
SELECT COUNT(customer_id)
FROM customers;
```

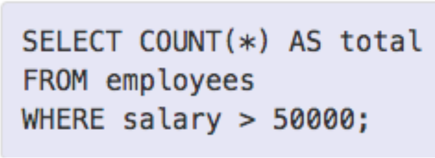


COUNT(customer_id)
6

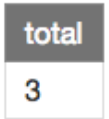
Example 2:



employee_number	last_name	first_name	salary	dept_id
1001	Smith	John	62000	500
1002	Anderson	Jane	57500	500
1003	Everest	Brad	71000	501
1004	Horvath	Jack	42000	501



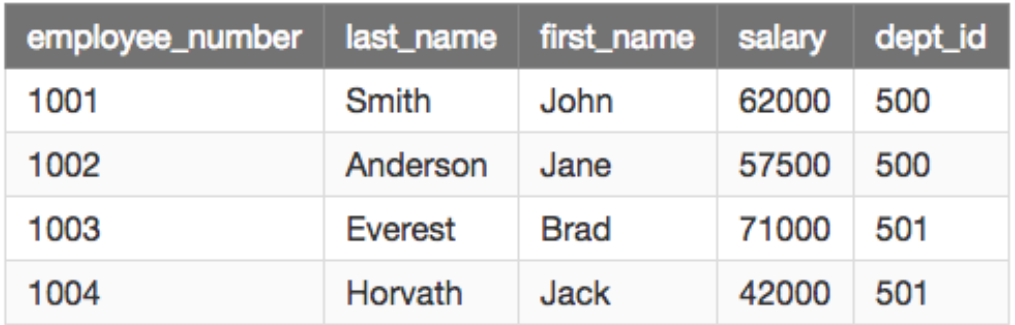
```
SELECT COUNT(*) AS total
FROM employees
WHERE salary > 50000;
```



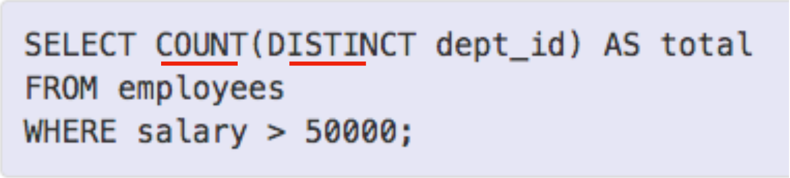
total
3

Count & Distinct

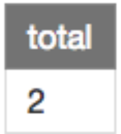
Example 3:



employee_number	last_name	first_name	salary	dept_id
1001	Smith	John	62000	500
1002	Anderson	Jane	57500	500
1003	Everest	Brad	71000	501
1004	Horvath	Jack	42000	501



```
SELECT COUNT(DISTINCT dept_id) AS total
FROM employees
WHERE salary > 50000;
```



total
2

Select:
SUM, GROUP BY, TOP

Example 1:

```
SELECT Status, SUM(AGE) FROM Customers GROUP BY (Status);
```

Output 1:

Status	SUM (Age)
S	91
M	77
D	45

Example 2:

```
SELECT TOP 2 Status, SUM(AGE) FROM Customers GROUP BY (Status);
```

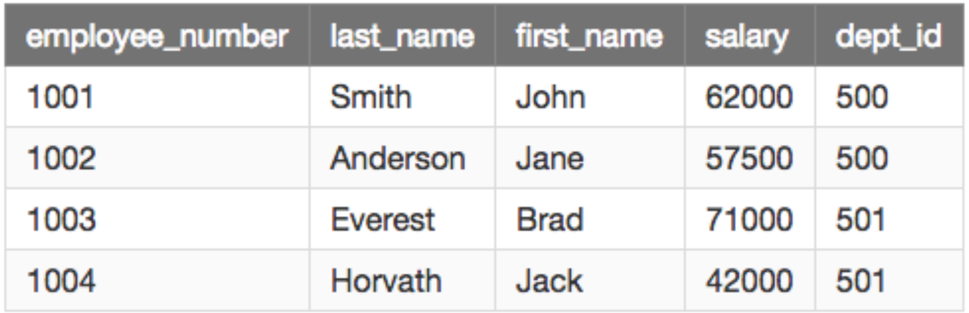
Output 2:

Status	SUM (Age)
S	91
M	77

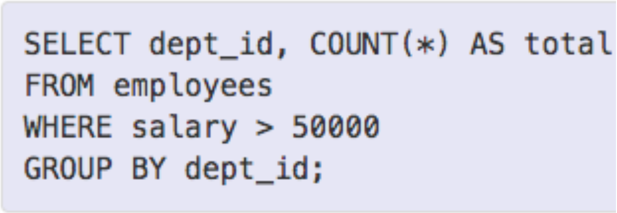
UNIVERSITÉ DE GENÈVE
CENTRE UNIVERSITAIRE D'INFORMATIQUE

Select- Group by

Example 3:

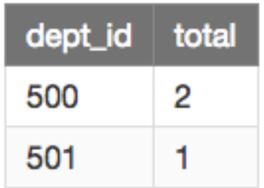


employee_number	last_name	first_name	salary	dept_id
1001	Smith	John	62000	500
1002	Anderson	Jane	57500	500
1003	Everest	Brad	71000	501
1004	Horvath	Jack	42000	501



```
SELECT dept_id, COUNT(*) AS total
FROM employees
WHERE salary > 50000
GROUP BY dept_id;
```

Output 3:



dept_id	total
500	2
501	1

SUM Function

Syntax:

```
SELECT expression1, expression2, ... expression_n,
       SUM(aggregate_expression)
FROM tables
[WHERE conditions]
GROUP BY expression1, expression2, ... expression_n;
```

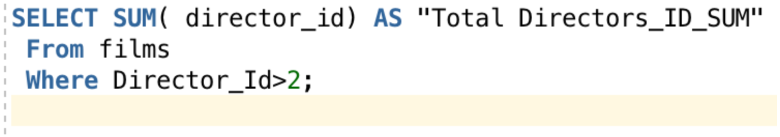
Example 1:

```
SELECT SUM (Age) FROM Customers;
```

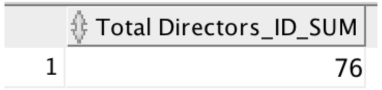
SUM (Age)
213

id	Name	Canton	Sex	Age	Status
0	Patrick	GE	2	24	S
1	Marc	BE	2	34	M
2	Sabrina	VD	1	45	D
3	Ivan	ZG	3	67	S
4	Arianna	VS	1	43	M

Example 2:

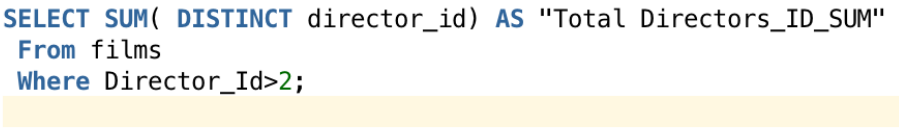


```
SELECT SUM( director_id) AS "Total Directors_ID_SUM"
 From films
 Where Director_Id>2;
```

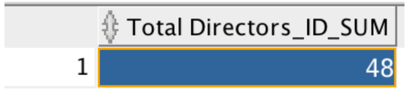


	Total Directors_ID_SUM
1	76

Example 3:



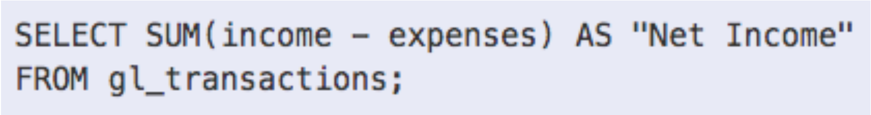
```
SELECT SUM( DISTINCT director_id) AS "Total Directors_ID_SUM"
 From films
 Where Director_Id>2;
```



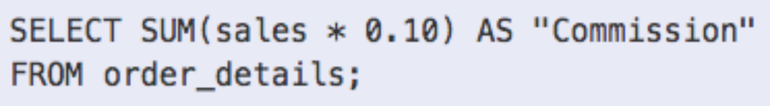
	Total Directors_ID_SUM
1	48

SUM Function

other examples:

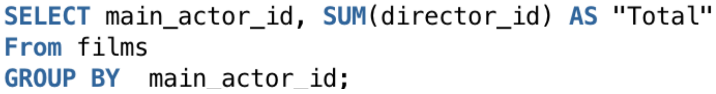


```
SELECT SUM(income - expenses) AS "Net Income"
FROM gl_transactions;
```

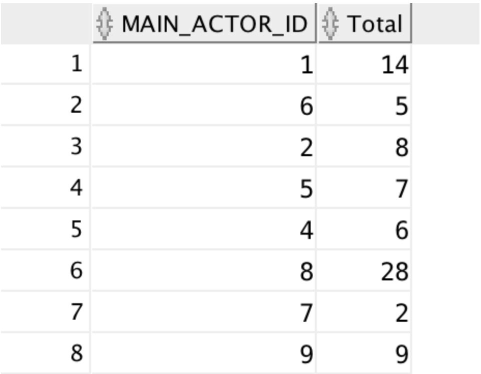


```
SELECT SUM(sales * 0.10) AS "Commission"
FROM order_details;
```

SUM function & GROUP BY:



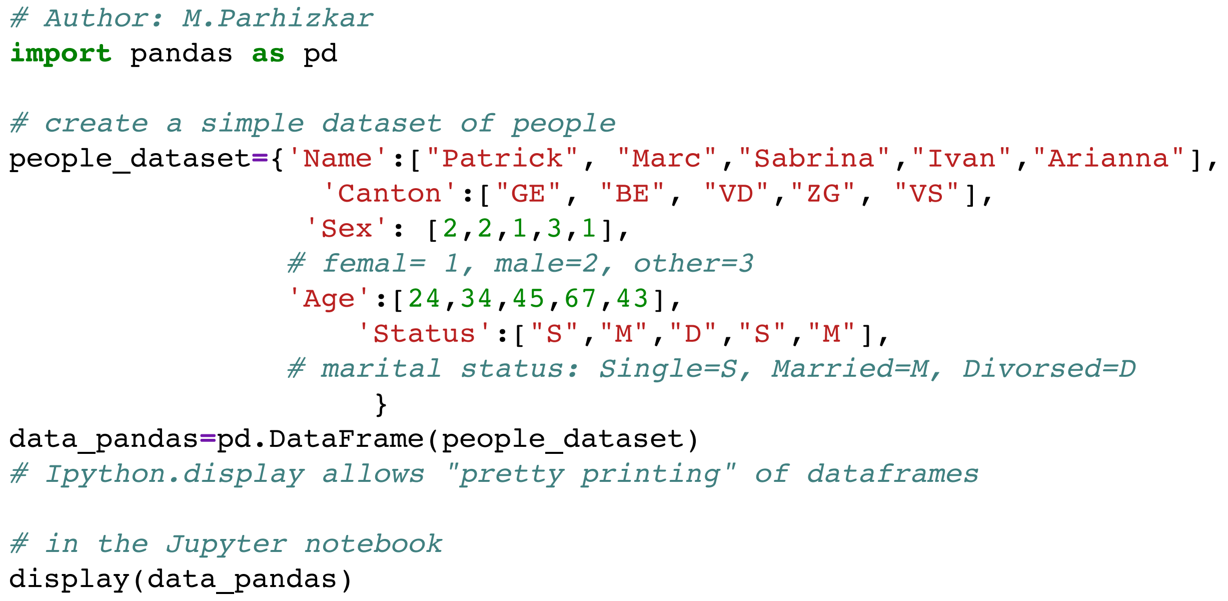
```
SELECT main_actor_id, SUM(director_id) AS "Total"
From films
GROUP BY  main_actor_id;
```



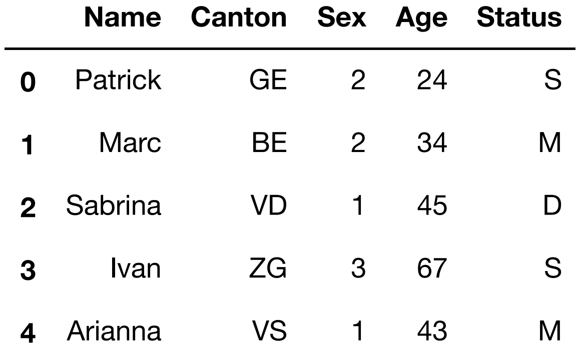
	MAIN_ACTOR_ID	Total
1	1	14
2	6	5
3	2	8
4	5	7
5	4	6
6	8	28
7	7	2
8	9	9

Application example…!
Where do we use database queries syntax?

Python- *pandas*

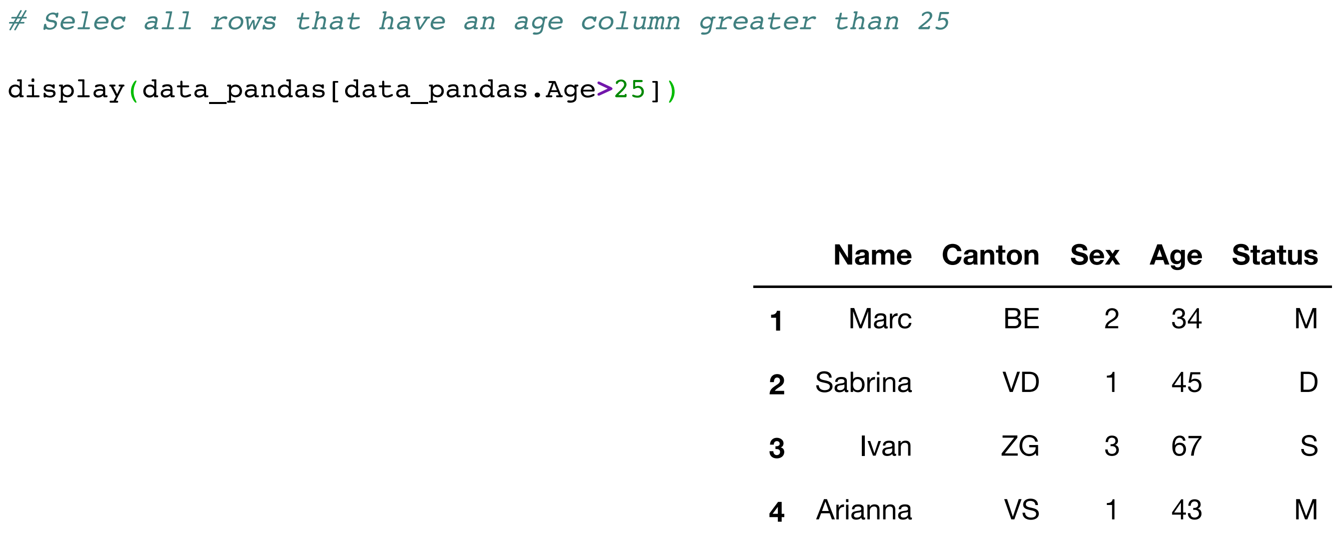


```
# Author: M.Parhizkar
import pandas as pd

# create a simple dataset of people
people_dataset={'Name':["Patrick", "Marc","Sabrina","Ivan","Arianna"],
                  'Canton':["GE", "BE", "VD","ZG", "VS"],
                 'Sex': [2,2,1,3,1],
                # femal= 1, male=2, other=3
                'Age':[24,34,45,67,43],
                    'Status':["S","M","D","S","M"],
                # marital status: Single=S, Married=M, Divorsed=D
                     }
data_pandas=pd.DataFrame(people_dataset)
# Ipython.display allows "pretty printing" of dataframes

# in the Jupyter notebook
display(data_pandas)
```



	Name	Canton	Sex	Age	Status
0	Patrick	GE	2	24	S
1	Marc	BE	2	34	M
2	Sabrina	VD	1	45	D
3	Ivan	ZG	3	67	S
4	Arianna	VS	1	43	M

Python- *pandas*



```
# Selec all rows that have an age column greater than 25

display(data_pandas[data_pandas.Age>25])
```

	Name	Canton	Sex	Age	Status
1	Marc	BE	2	34	M
2	Sabrina	VD	1	45	D
3	Ivan	ZG	3	67	S
4	Arianna	VS	1	43	M

Thank you!

For the next sessions, we start at 08:30, non-stop until 10:00