Vilnius
University

Vilnius University

D1/02. Structured Query Language (SQL)

School on the Database Infrastructure for the CMS Phase 2 Upgrade

Outline

Vilnius University

- SQL
- SQLPlus CLI
  - Hands-on
- Data Definition Language (DDL)
  - Create + Alter + Drop
  - Hands-on
- Data Manipulation Language (DML)
  - Insert + Update + Delete
  - Hands-on
- Queries
  - Select, Join + Projection + Filter
  - Hands-on

Structured Query Language (SQL)

Vilnius University

- SQL is the standard language for relational database management systems
- We can use SQL in at least 2 different ways:
  - Interactively - write queries and access the database directly and get results immediately
  - Embedded - use SQL with other program, which use different language than SQL
- SQL categories
  - Data definition (Data Definition Language, or DDL)
  - Data manipulation (Data Manipulation Language, or DML)
  - Querying data (Data retrieval)
- With SQL we can do basic functions of persistence storage
  - **C**reate
  - **R**ead
  - **U**pdate
  - **D**elete

SQL History

Vilnius University

- History
  - Raymond F. Boyce and Donald D. Chamberlin started the initial development of SQL at IBM in the beginning of the 1970s.
- Standards
  - By 1986, ANSI and ISO standard groups officially adopted the standard "Database Language SQL" language definition. New versions of the standard were published in 1989, 1992, 1996, 1999, 2003, 2006, 2008, 2011, 2016.
  - DBMS specific
    - Oracle
    - MySQL
    - MariaDB
    - PostgreSQL
    - ..

SQL Query

Vilnius University

- SQL Query:

  SELECT <attributes>
  FROM <one or more relations>
  WHERE <conditions>

- SQL Query example:

  SELECT *table_name*
  FROM *user_tables*;

SQL Plus

Vilnius University

SQL Plus - interactive command-line (CLI) tool that provides access to the Oracle database.

- Sql Plus enables:
  - Enter SQL Plus commands to configure the SQL Plus environment
  - Startup and Shutdown an Oracle database
  - Connect to an Oracle database
  - Enter and execute SQL commands and PL/SQL blocks
  - Format and print query results

Task 0 - Connect To Database

Vilnius University

1. ssh -L 1525:dbschool-srv01:1521 username@lxplus.cern.ch
2. Connect to Database with your details.

- sqlplus CMS_{lastName}/{firstName}@dbschool-srv01:1521/xe
- Sqlplus64 CMS_{lastName}/{firstName}@dbschool-srv01:1521/xe

2. Prepare environment

- Cheats Twiki
- Useful SQL PLUS commands:
  - DEFINE _EDITOR to choose your editor
    - DEFINE _EDITOR=vi
    - DEFINE _EDITOR=nano
    - DEFINE _EDITOR=vim
  - *$ ed (edit)* - allows you edit previous command
  - *$ /* - execute previous command from buffer (can be changed by “edit”)
  - $ desc - describe table
    - e.g. desc Parts;
- e.g. CMS_SILALE/onuskis@dbschool-srv01:1521/xe

Data visualization

Vilnius University

- Prepare your terminal for better data visualization
  - set termout off
  - set verify off
  - set trimspool on
  - set linesize 200
  - set longchunksize 200000
  - set long 200000
  - set pagesize 1000
  - Set termout on

Data Definition Language (DDL)

Vilnius University

- Data Definition Language - allows you to create, modify and remove components of a database (DB).
- There are many DB structure components like
  - Tables
  - Views
  - Sequences
  - Constraints
  - Indexes
  - ...
- Possibility to change structure of Table without changing everything. E.g. Add new column to table, while other users are using DB.

Data types

Vilnius University

- In SQL each table’s columns are meant to hold a specific type of data
- There are more than 30 different data types
- 3 the most used data types
  - NUMBER
  - VARCHAR
  - DATETIME

Data types (NUMBER/VARCHAR/DATETIME)

Vilnius University

- Number - Integer Data Type
  - An NUMBER datatype can store number values up to 38 significant digits
- VARCHAR/NVARCHAR - Text Values
  - VARCHAR can store maximum 8000 characters
    - If we defined CityName(VARCHAR30) but CityName was “Geneva” (6 letters) it takes only 6 characters space.
  - NVARCHAR can store UNICODE and as UNICODE occupy twice the space, NVARCHAR can store MAXIMUM 4000 characters.
- DATETIME - DATE and TIME
  - USED to store the date and time

Data Types DEMO

Vilnius University

```
CREATE TABLE TableName (
        "ID" NUMBER(32) NOT NULL,
        "NAME" VARCHAR(40),
         "CITY"  VARCHAR(40),
        PRIMARY KEY (ID)
)
```

Task 1 - Create Tables

Vilnius University

1. Analyse ER diagram and create KIND_OF_PARTS and PARTS tables

HINT:

```
CREATE TABLE TableName (
        "ID" NUMBER(32) NOT NULL,
        "NAME" VARCHAR(40),
        "CITY"  VARCHAR(40),
        PRIMARY KEY (ID)
)
```

Barcode - 1235436
SerialNumber - A123B32C

Task 1 - Solution

Vilnius University

```
CREATE TABLE KIND_OF_PARTS (
        "ID" NUMBER(32) NOT NULL,
        "NAME" VARCHAR(40) NOT NULL,
        PRIMARY KEY (ID)
)


CREATE TABLE PARTS (
        "ID" NUMBER(32) NOT NULL,
        "SERIAL_NUMBER" VARCHAR(32) NOT NULL,
        "BARCODE" NUMBER(32) NOT NULL,
        PRIMARY KEY (ID)
)
```

Alter Table

Vilnius University

- Then table is created there is possibility to modify it without dropping and recreating
- We can not combine two different commands e.g. Drop one column and add another at the same time
- Possible Changes
  - Add/Modify/Rename/Drop Column
  - Add/Drop Index
  - Add/Drop Constraint

Alter Table Demo

Vilnius University

- Add additional column:

  ALTER TABLE TableName
  ADD ColumnName NUMBER(32)

- Remove one column:

  ALTER TABLE TableName
  DROP COLUMN ColumnName

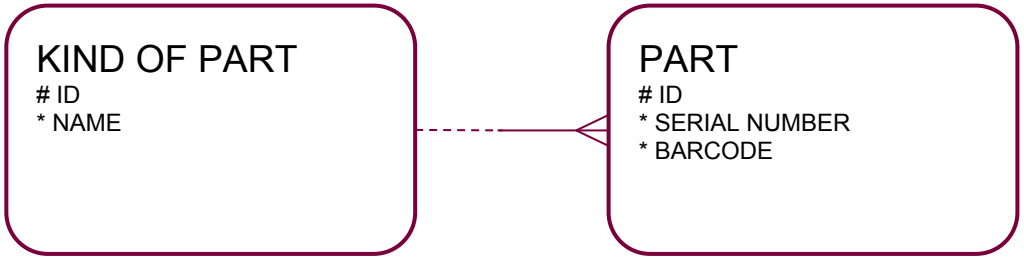
- Add Foreign Key:

  ALTER TABLE TableName
  ADD CONSTRAINT CONSTRAINT_NAME
  FOREIGN KEY (Table_Column)
  REFERENCES TableName(Table_Column)

TASK 2 - Add additional Column & FK

Vilnius University

1. Add Column “KOP_ID” to PARTS table
2. Create FK from PARTS.KOP_ID to KIND_OF_PARTS.ID



HINT

ALTER TABLE TableName
ADD ColumnName NUMBER(32)

ALTER TABLE TableName
ADD CONSTRAINT CONSTRAINT_NAME
FOREIGN KEY (Table_Column) REFERENCES
TableName(Table_Column)

Task 2 - Solution

Vilnius University

1. Add additional Column:

```
ALTER TABLE PARTS
ADD KOP_ID NUMBER(32)
```

2. Create Foreign key

```
ALTER TABLE PARTS
ADD CONSTRAINT FK_KOP_ID
FOREIGN KEY (KOP_ID) REFERENCES KIND_OF_PARTS(ID)
```

Sequences

Vilnius University

- A sequence - generated integers used for primary keys.

  CREATE SEQUENCE **sequence_name**

  [START WITH **start_num**]

  [INCREMENT BY **increment_num**] [

  { MAXVALUE **maximum_num** | NOMAXVALUE }]

  [{ MINVALUE **minimum_num** | NOMINVALUE }];

Sequence Demo

Vilnius University

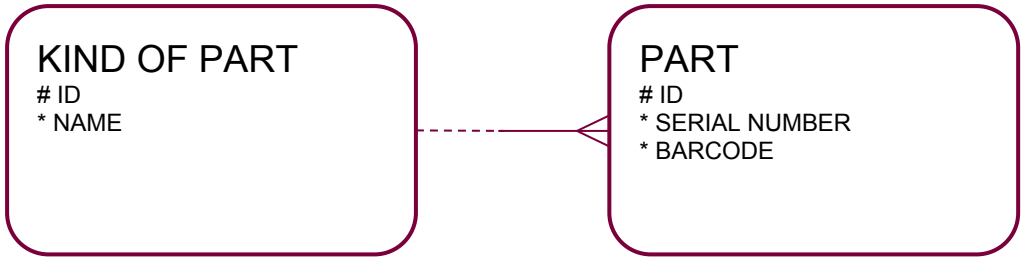
- Sequence

CREATE SEQUENCE seq_name
START WITH value
INCREMENT BY value;

TASK 3 - Create sequences

Vilnius University

1. Create sequences
   a. KIND_OF_PARTS.ID (with name - *seq_kop)*
   b. PARTS.ID (with name - seq_part)



HINT

CREATE SEQUENCE seq_name
START WITH value
INCREMENT BY value;

TASK 3 - Solution

Vilnius University

1. Create sequences
   1.1.
   ```
   CREATE SEQUENCE seq_kop
   START WITH 1
   INCREMENT BY 1;
   ```
   1.2.
   ```
   CREATE SEQUENCE seq_part
   START WITH 1
   INCREMENT BY 1;
   ```

DB objects from file

Vilnius University

- We can create DDL file with extension .sql with all db objects
- In order to load tables with all constraints use these commands:
  - @{file}
  - @{path}{file}
  - @/scripts/script.sql

Task 4 - execute DDL file

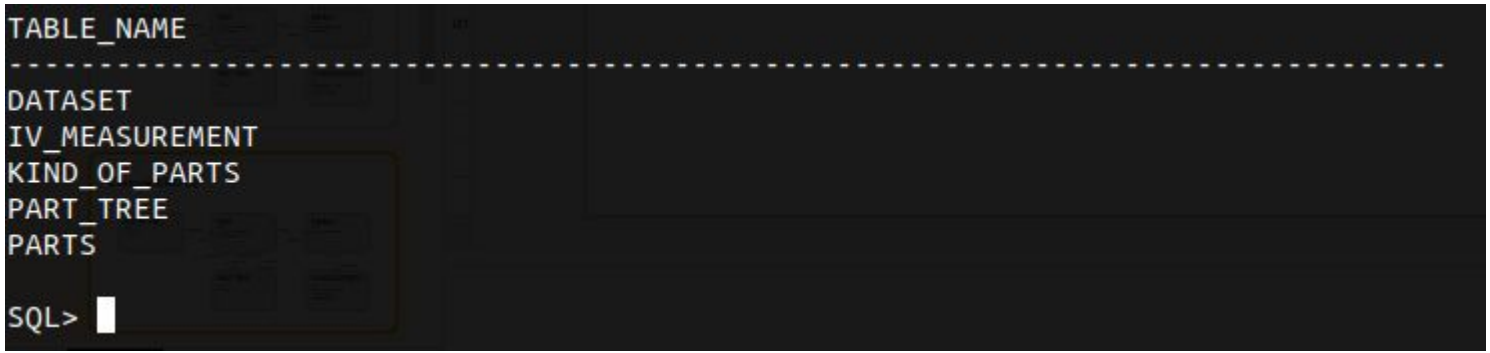
Vilnius University

- Path: */afs/cern.ch/user/v/valdo/public/dbschool/*
- *WEB Path: http://valdo.web.cern.ch/dbschool/*
- File name : script.sql
- Execute script and create remaining tables

Task 4 -Solution

Vilnius University

- *Command to execute script with path*
  - *$ @/afs/cern.ch/user/v/valdo/public/dbschool/script.sql*
  - *http://valdo.web.cern.ch/valdo/dbschool/*
- *Execute this line in order to see all your tables:*
  - *SELECT table_name FROM user_tables;*
- *You should see:*



Data Manipulation Language (DML)

Vilnius University

- DML allows to change the content of database.
- 3 basic data manipulation commands:
    - Insert (Add rows to a table)
    - Update (Change column values of existing rows)
    - Delete (remove rows from a table)

Data Insertion

Vilnius University

- The Oracle INSERT statement is used to insert a single record or multiple records into a table in Oracle.
- Data insertion

INSERT INTO table_name VALUES (column1, column2,...)
VALUES (value1, value2);

Task 5 - Insert data

Vilnius University

1. Insert data into KIND_OF_PART TABLE
   a. ID from seq_kop.nextval
   b. Name - “Hamamatsu Sensor”
2. Insert data into KIND_OF_PART TABLE
   a. ID from seq_kop.nextval
   b. Name - “Test”

HINT

INSERT INTO table_name VALUES (column1, column2,...)

VALUES (value1, ‘test’);

Task 5 - Solution

Vilnius University

1. Insert data into KIND_OF_PARTS TABLE
   a.
   ```
   INSERT INTO KIND_OF_PARTS (ID, NAME)
   VALUES (seq_kop.nextval, 'Hamamatsu Sensor');
   ```
   b.
   ```
   INSERT INTO KIND_OF_PARTS (ID, NAME)
   VALUES (seq_kop.nextval, 'Test');
   ```

Update existing data

Vilnius University

- The UPDATE statement is used to update existing records in a table
  - Syntax

```
UPDATE table_name
SET column1 = expression1, ... column_n = expression_n
 [WHERE conditions];
```

  - Example

```
UPDATE Parts
SET  serial_number = new_serial
WHERE id = 1;
```

Task 6 - Update table row

Vilnius University

1. Update KOP table row, where Name='Test'
   a. Set Name='updated'

HINT

```
UPDATE Parts
SET  serial_number = new_serial
WHERE id = 1;
```

Task 6 - Solution

Vilnius University

1. Update row

```
update kind_of_parts
set name='updated'
where name='test'
```

Delete row from table

Vilnius University

- The DELETE statement is used to delete a single record or multiple records from a table in Oracle
  - Syntax

    DELETE FROM table
    [WHERE conditions];

  - Example

    DELETE FROM Parts
    WHERE barcode = 1273;

Task 7 - Delete record from Table

Vilnius University

- Delete record from KIND_OF_PARTS Table
  - Where name=’updated’

HINT

DELETE FROM Parts
WHERE barcode = 1273;

Task 7 - Solution

Vilnius University

1. Delete record form kind_of_parts

```
Delete from KIND_OF_PARTS
WHERE name = 'updated'
```

Upload all data from Script

Vilnius University

- *Command to execute script with path*
  - *$ @/afs/cern.ch/user/v/valdo/public/dbschool/data.sql*
- *This command fill all tables with data for next tasks*

Data Retrieval

Vilnius University

- There are many ways how you can select data from Database.
- SELECT statement is very complex and consist of many clauses like:
  - Group
  - Sort
  - Join
  - Order
  - Having
- We will cover some of them

Simplest Select Statement with Ordering

Vilnius University

- Order by statement used to sort the result-set in ascending or descending order
- Structure of Select statement with ordering is pretty simple:

SELECT expressions
FROM tables
[WHERE conditions]
ORDER BY [Column];

- To select all data from **PARTS** just execute command below:

*SELECT * FROM PARTS ORDER BY KOP_ID DESC;*

Joins

Vilnius University

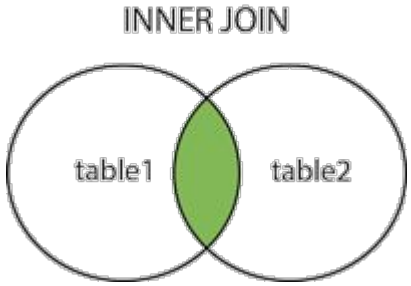
- A JOIN clause is used to combine rows from two or more tables, based on a related column between them
- 5 Types of joins
  - Left
  - **Right**
  - **Inner**
  - **Outer**
  - Full

Inner Join

Vilnius University

- The INNER JOIN keyword selects all rows from both tables as long as there is a match between the columns
- Syntax

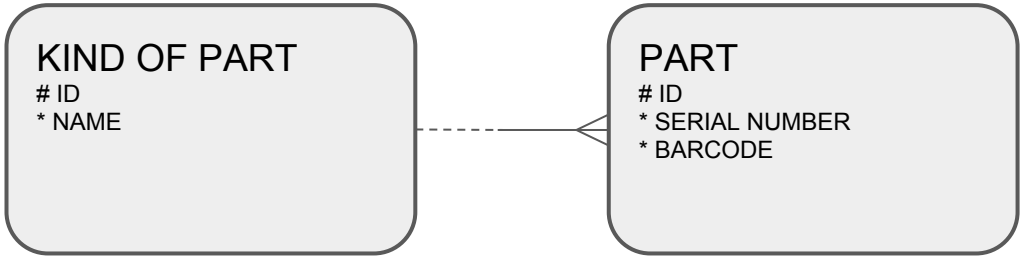
```
SELECT table1.column, table2.column
From table1
INNER JOIN table2
ON table1.relatedColumn = table2.relatedColumn
```



Task 8 - Inner join

Vilnius University

- Select
  - PARTS.SERIAL_NUMBER
  - PARTS.BARCODE
  - KINDS_OF_PARTS.NAME
  - 3 columns:
    - Name
    - Serial number
    - barcode



Task 8 - Solution

Vilnius University

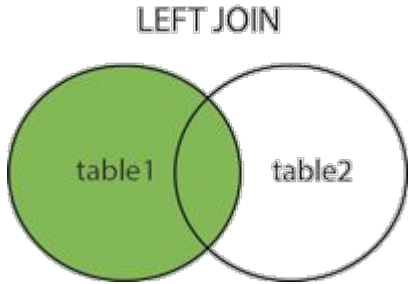
```
SELECT KIND_OF_PARTS.name, PARTS.serial_number, PARTS.barcode
FROM Parts
INNER JOIN KIND_OF_PARTS ON PARTS.kop_id =KIND_OF_PARTS.id;
```

Left Join

Vilnius University

- The LEFT JOIN keyword returns all records from the left table (table1), and the matched records from the right table (table2).
- Syntax

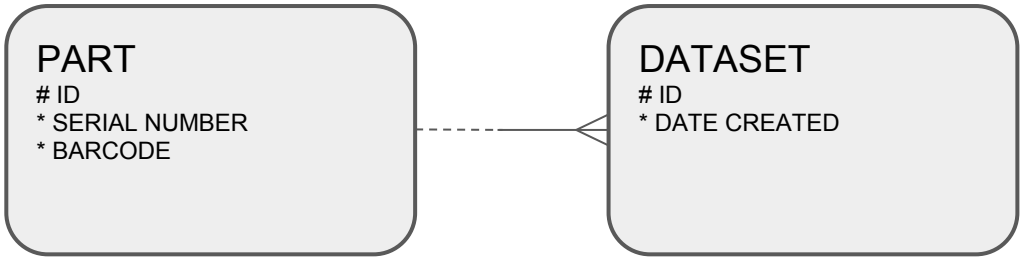
```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```



Task 9 - Left Join

Vilnius University

- Select
  - PARTS.SERIAL_NUMBER
  - DATASETS.DATE_CREATED



HINT

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

Task 9 - Solution

Vilnius University

```
SELECT PARTS.SERIAL_NUMBER, DATASET.DATE_CREATED
FROM PARTS
LEFT JOIN DATASET
ON PARTS.ID = DATASET.PART_ID;
```

Sub Queries

Vilnius University

- A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.
- Syntax

```
SELECT column_name [, column_name ]
FROM table1 [, table2 ]
WHERE column_name OPERATOR
        (SELECT column_name [, column_name ]
        FROM table1 [, table2 ]
        [WHERE])
```

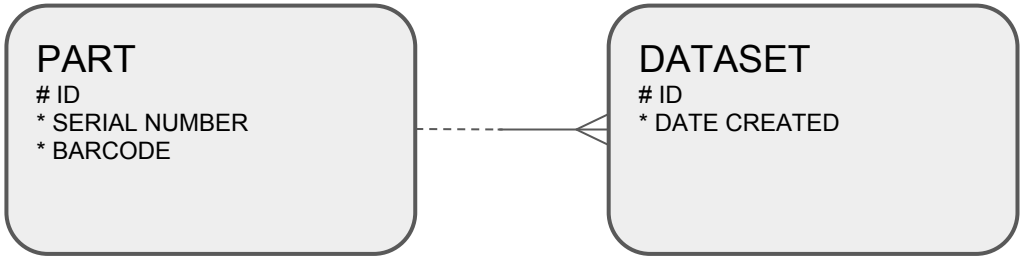
Task 10 - Sub queries

Vilnius University

- SELECT
  - PARTS.SERIAL_NUMBER
- Condition
  - DATASET.DATE_CREATED > 2018-08-15
- Use Sub Query with operator IN

Hint

SELECT PARTS.SERIAL_NUMBER
FROM PARTS
WHERE PARTS.ID IN
(SELECT SUB QUERY);



Task 10 - Solution

Vilnius University

```
SELECT PARTS.SERIAL_NUMBER
FROM PARTS
WHERE PARTS.ID IN
(SELECT PART_ID FROM DATASET WHERE DATE_CREATED > '2018-08-15');
```

GROUP BY

Vilnius University

- Group by statement is often used with aggregate functions (count, max, min) to group the result-set by one or few columns

GROUP BY

Vilnius University

- Mixed statement to find out maximum current_amps for unique ID from IV_MEASUREMENTS

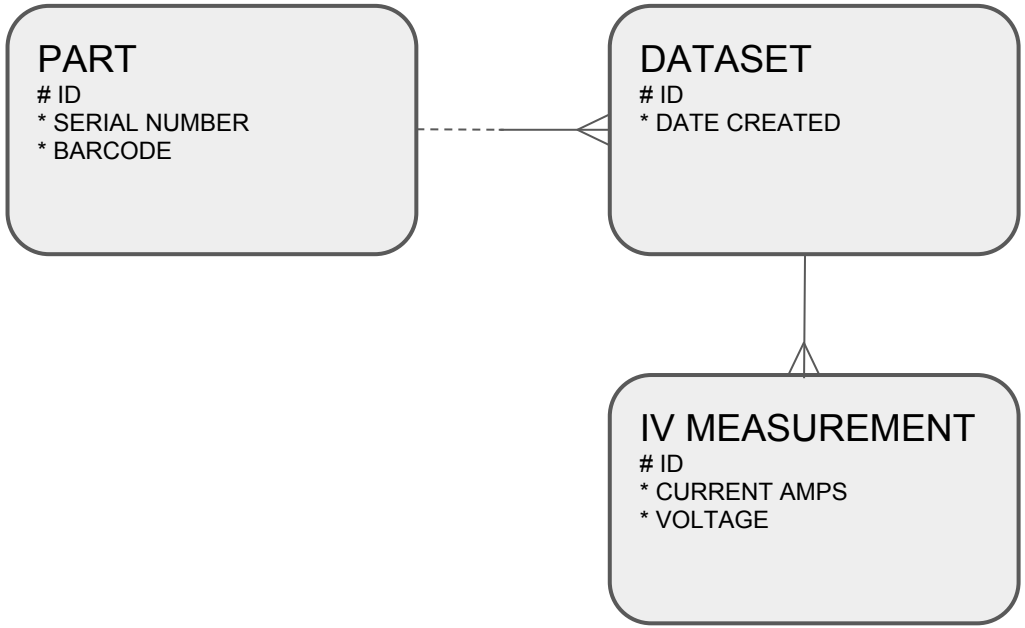
```
Select max(current_amps), dataset_id
FROM IV_MEASUREMENTS
GROUP BY dataset_id
order by dataset_id;
```

	ID	CURRENT_AMPS	VOLTAGE	DATASET_ID
1	1	399	180	1
2	2	400	200	1
3	3	401	220	1
4	4	402	240	1
5	5	404	260	1
6	6	22	0	2
7	7	260	20	2
8	8	329	40	2
9	9	369	60	2
10	10	386	80	2

TASK 11 - Mixed Task

Vilnius University

- Use COUNT() function
- Print Part name and number of measurements
- Use 2 inner joins



HINT

SELECT PARTS.id, COUNT(dataset_ID)
FROM PARTS
INNER JOIN …….
INNER JOIN ……
Group by parts.id
Order by parts.id;

Task 11 - Solution

Vilnius University

```
SELECT PARTS.id, COUNT(dataset_ID)
FROM PARTS
INNER JOIN DATASET on PARTS.id = DATASET.part_id
INNER JOIN IV_MEASUREMENT ON DATASET.id = IV_MEASUREMENT.dataset_id
Group by parts.id
Order by parts.id;
```

View

Vilnius University

- View - is a logical table based on one or more tables or views.
- A view contains no data itself.
- The tables upon which a view is based are called base tables.

Syntax

```
CREATE VIEW view_name AS
SELECT columns
FROM tables
WHERE conditions;
```

Task 12 - Create view

Vilnius University

- Use “ed” command and edit your previous script to create view with the same functionality
- View name should be - partsCountView

Hint

```
CREATE VIEW view_name(columns) AS
SELECT columns
FROM tables
WHERE conditions;
```

Task 12 - Solution

Vilnius University

```
CREATE VIEW testView (part_id,num_of_measurements) AS
SELECT PARTS.id, COUNT(dataset_ID)
FROM PARTS
INNER JOIN DATASET on PARTS.id = DATASET.part_id
INNER JOIN IV_MEASUREMENT ON DATASET.id = IV_MEASUREMENT.dataset_id
Group by parts.id
Order by parts.id;


Select * from testView;
```

Summary

Vilnius University

- It is not so easy to understand SQL
- It has many ways to create, edit Tables, views
- It has possibility to run commands from files to
  - Create Tables and all DB objects
  - Insert/edit/delete data in tables
- It has possibility to get various data from all tables, views
  - Joins, groups by, order, sub-queries


VILNIAUS UNIVERSITETAS
1579
UNIVERSITAS VILNENSIS
440