# Assignment 1 - Views

Let’s revisit the company example.

1. Create a SQL statement which creates the view V1. View V1 has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project.

CREATE VIEW ASS1 AS

SELECT p.pname AS PROJECT\_NAME, d.dname AS CONTR\_DEP, count(distinct w.eSSN) AS NUM\_EMP, SUM(w.hours) AS HOURS\_PER\_WEEK

FROM project p, department d, works\_on w

WHERE d.dnumber = p.dnum and w.pno = p.pnumber

GROUP BY p.pname;

1. Create a SQL statement which creates the view V2. View V2 that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project with more than one employee working on it.

CREATE VIEW ASS1 AS

SELECT p.pname AS PROJECT\_NAME, d.dname AS CONTR\_DEP, count(distinct w.eSSN) AS NUM\_EMP, SUM(w.hours) AS HOURS\_PER\_WEEK

FROM project p, department d, works\_on w

WHERE d.dnumber = p.dnum and w.pno = p.pnumber

GROUP BY p.pname

HAVING count(distinct w.eSSN) > 1;

1. Assume a view with following code is given.

**CREATE VIEW** seniors **AS**

**SELECT** \*   
**FROM** employee   
**WHERE** salary > 45000

**WITH CHECK OPTION**;

* Explain the purpose of views in general and the function of the view seniors in particular.

Views are interfaces that provide specific information for certain user groups or programs.   
This leads to a higher robustness of these programs against changes in the database.  
Moreover, it’s necessary to exclude or combine certain information when needed (e.g. security, permission handling)

**Seniors** shows all information about employees that have a higher salary then 45 000.

* Is it possible to modify the given view seniors?
  + First think about it in theory.
  + Second try it out in a database:
    - Update the salary of one employee in view seniors and in base table employee.
    - Insert some new employee in view seniors and in base table employee.
    - Delete some tuples in view seniors and in base table employee.

Seniors is a selection-view that may have problems with tuple migration. The newer databases allow modification and only forbid them if tuple migration occurs.

# Assignment 2 - Transactions

For this assignment, you need two distinct sessions (connections) to your database.

* With Oracle this can be achieved by two unshared connections in SQL Developer (an unshared connection can be opened by typing Ctrl-Shift-N in the SQL pane).   
  Alternatively, you can use two instances accessing the same database schema (log on as the same user).
* In MySQL Workbench you might have to allow opening multiple instances first. On Windows this can be achieved via Edit 🡪 Preferences 🡪 General (tab) 🡪 Others (section) 🡪 Allow more than one instance of MySQL Workbench to run.

The two database sessions are denoted as session1 and session2, resp.

For every task include your statements and the corresponding output in session1 and session2 as a table. Make sure everything is in chronological order.

Example:

|  |  |  |
| --- | --- | --- |
| Timestamp | Session 1 | Session 2 |
| 0 | INSERT INTO tab1 ...;  🡪 result screenshot  SELECT \* FROM tab1;  🡪 result screenshot | - |
| 1 |  | SELECT \* FROM tab1;  🡪 result screenshot |

1. In session1 run a command creating a table named ‘tab1’ with two attributes:
   * id with data type integer, primary key
   * n with data type integer

When is table ‘tab1’ visible in session2?

CREATE TABLE tab1(

id integer PRIMARY KEY,

n integer);

It is visible right away.

1. In session1 insert the following tuples into ‘tab1’ within one transaction:  
   (1,1), (2,2), (3,3)

What content of ‘tab1’ is displayed in session2 before and after you commit your changes in session1?

INSERT INTO tab1 VALUES(1,1);

INSERT INTO tab1 VALUES(2,2);

INSERT INTO tab1 VALUES(3,3);

BEFORE: none  
AFTER: 1,1 / 2,2 / 3,3

1. In session1 update the value of n to 33 for the tuple with id 3 (without commiting). Afterwards rollback that transaction.

What value of n (id=3) is displayed in session1 before and after the rollback?

What value of n (id=3) is displayed in session2 before and after the rollback?

UPDATE tab1 SET n = 33 WHERE id = 3;

BEFORE: 33  
AFTER: 3

BEFORE: 3  
AFTER: 3