# This hand in 3 will be updated later in wiseflow …

# Exercise 1: Views

1. Create a simple view for ‘works\_on’ table in the company database – all turples should be in this veiw!
2. Create a view with a sum of hours for each project.
3. Create a view with a sum of hours for each combination of employee and project, add names for employee and project and calculate the cost – the cost for each hour are 300 DDK.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EMP#** | **EMP\_Name** | **PROJ#,** | **Project Name** | **Hours** | **Cost** |
| 123456789 | Jens Jensen | 5 | Test | 8 | 2400 |
|  |  |  |  |  |  |

1. A view that has the department name, manager name and managers salary for every department.
2. A view that has the employee name, supervisor name, and employee salary for each employee who works in ‘Research’ department.
3. A view that has the project name, controlling department name, number of employees, and total hours worked per week on project for each project
4. A view 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.
5. A view that retrieve the names of all employees whose supervisor’s supervisor has ‘8886655555’ for Ssn.
6. A view that retrieve: for each department whose average employee salary is more than $30.000 retrieve the department name and the number of employees working for that department.
7. Create a view of you own choice.

(It could be a view who contain the following information …

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **EMP#** | **EMP\_Name** | **PROJ#,** | **Project Name** | **Hours** | **Cost** | **Total Cost** |
| 123456789 | Jens Jensen | 5 | Test | 8 | 2400 | 16000 |
| ... | ... | ... | ... | ... | ... | 16000 |
| ... | ... | ... | ... | ... | ... | 16000 |

)

# Exercise 2: Trigger

1. Create a log trigger for the ‘works\_on’ table in the company database – for insert, update and delete. The new log table should contain a serial id and a time stamp.

1. Create a trigger for table ‘project’ who raise an exception if a department assign more than 3 projects.
2. Create a trigger for the ‘works\_on’ table who raise an exception if the employee is assigned more than 4 projects.
3. Create a log trigger for department table in the company database …the content in the log table should be readable!
4. Create a trigger of your own choice.

# Exercise 3: JDBC

1. Create a database table in JAVA
2. Insert records to the database from JAVA
3. Update some of the records from JAVA
4. Delete the contents in the database table from JAVA
5. Drop the table again from JAVA
6. Documentation: only the relevant part from JAVA and PostgreSQL.

# Exercise 4: Transaction

1. Check the log file from exercise 1.
2. Explain ”dirty read problem” with data from first poster.
3. Explain ”non-repeatable read” with data from first poster.
4. Explain ”phantom read” with data from first poster.

# Exercise 5: Reverse Engineering:

From the ‘Invoice’ below, show the normalisation process for creating a relation database model:

Contractor Holmsen Aps

Vesterhavsvej 25

9990 Fnatting

# Invoice no 12

## Research number. 200207 Research date: 02/01/2016

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Analysis no.: | Description: | Amount | Price | Total |
| 10 | Pesticide Test | 2 | 1.200,00 | 2.400.00 |
| 15 | Bacteria Analysis | 3 | 500,00 | 1.500,00 |
|  |  |  |  |  |
|  |  | Total excl. VAT | | 3.900,00 |
|  |  | VAT | | 975,00 |
|  |  | Total incl. VAT. | | 4.875,00 |

ØKOLOGIC

Chr.Østergårds Vej 10

8700 Horsens

# Exercise 6: Joins

In this exercise, you have to use the following types of 'join':

Union, intersect, except, natural join, inner join, left join, right join.

Given the following tables:

List of payments from your bank:

|  |  |  |
| --- | --- | --- |
| Invoice Number | Customer | Value |
| 123 | Peter | 200 |
| 234 | Soren | 500 |
| 345 | Soren | 400 |
| 456 | Peter | 66 |
| 567 | Trine | 50 |
| 1212 | Niels | 87 |
| 1313 | Viggo | 99 |

List of debtors ... customers who should have paid:

|  |  |  |
| --- | --- | --- |
| Invoice Number | Customer | Value |
| 012 | Hans | 600 |
| 123 | Peter | 200 |
| 234 | Soren | 500 |
| 345 | Soren | 400 |
| 456 | Peter | 66 |
| 567 | Trine | 50 |

Answer the following question:

* Make a list with invoices who have been paid.
* Make a list of invoices who have not been paid.
* Make a list of customer who do not have any invoice but have paid an invoice … maybe twice.

The same question can be answered by using various types of 'joins' - show these!