

## Experiment 4 Simple queries (2)

### 1. Experimental objectives

- 1) Master the use of GROUP BY and HAVING clauses.
- 2) Master the methods of modification of tables.

### 2. Experimental environment

SQL Server 2017

### 3. Experimental key points

- 1) Attaching the projemp database that will be used into the lab environment.

**projemp** relational schema:

**DEPT** (**dno**, **dname**, **location**)

**EMP** (**eno**, **ename**, **salary**, **age**, **supno**, **dno**\*)

**WORKS** (**eno**\*, **pno**\*, **role**)

**PROJ** (**pno**, **pname**, **ptype**, **budget**)

- 2) Find the table that should be used in the queries.
- 3) Use GROUP BY and HAVING clauses to complete the queries.
- 4) Use INSERT, DELETE and UPDATE clauses to modify the tables.

### 4. Experimental content

#### 4.1 Use of GROUP BY and HAVING statements

- 1) Get the average salary in each department (or by department)

**NOTE:** The attribute (**dno**) in the **SELECT** clause *MUST* agree with the **GROUP BY** attribute (**dno**).

The **HAVING** clause can *ONLY* be used when a **GROUP BY** clause is present and is used to choose between groups.

- 2) How many employees work on each project (using **WORKS** table)?
- 3) Get the departments having an average salary below £30,000.

**Note:** Only departments d2 and d3 are selected because their average salary is less than £30,000.

**HAVING** clauses choose between groups and *MUST* conform to the syntax:

<i>Aggregate function</i>	<i>compared with</i>	<i>a value</i>
(count( ), avg( ), etc)	(=, <, >, etc)	(number)

- 4) Get the projects having less than 9 employees working on them (i.e. **HAVING** a **COUNT** of employees less than 9)
- 5) Get a unique list of department numbers (Use of **DISTINCT** predicate - eliminates duplicate records from result)
- 6) How many employees are there in each department? (See **query 1**)
- 7) List the departments (by department number) that have a total salary of more than £300,000. (see **query 2**)
- 8) How many employees earn £20,000 or £25,000?

#### 4.2 INSERT, DELETE and UPDATE

- 1) Insert a new record with eno 'e51', ename 'mallon', age 32, salary 26000.00 , supno 'e17' and dno 'd2';

**Note:** Check the database to see if this record has been entered.

- 2) Delete the employee with an employee number of 'e51'.

**Note:** Check the database to see if this record has been deleted

- 3) Using similar SQL code, try to delete the employee with an employee number of 'e19', you should find that an error is generated. Study the error message and determine why the deletion operation is prevented. Ask a demonstrator if you are unsure.
- 4) Change the budget of project number 'p13' to £650,000 (is £520,000 at present).

**Note:** Check the database to see if this record has been changed

- 5) Change the employee named 'oliver' to 'pearson'.

- 6) Add a new record with the following values:

eno 'e60', ename 'young', age 51, salary 60000.00, supno 'e1',  
dno 'd1'

- 7) Now delete this record using the SQL DELETE command.