# JOINS

1. Create:
   1. Equijoin
   2. Natural join
   3. Cartesian product

for the **project** and **works\_on** tables(refer Assignment1).

**Ans:**

**a.** SELECT project\_no FROM project JOIN works\_on ON project.project\_no = works\_on.project\_no;

b. SELECT project\_no FROM project NATURAL JOIN works\_on ON project.project\_no = works\_on.project\_no;

c. SELECT project\_no FROM project CROSS JOIN works\_on ON project.project\_no = works\_on.project\_no;

1. Get the employee numbers and job titles of all employees working on project Gemini

**Ans** : SELECT emp\_no, Job FROM works\_on where project\_no in (SELECT proj\_no FROM project WHERE project\_name = 'Gemini');

1. Get the first and last names of all employees that work for departments *Research* or *Acounting.*

**Ans :** SELECT firstName, lastName FROM employee where employee\_no IN (SELECT employee\_no FROM works\_on where department\_no in (SELECT department\_no FROM department WHERE (department\_name = 'Research' or department\_name= 'Accounting')));

1. Get the enter dates of all clerks that belong to the department d1.

**Ans** : SELECT enterdates FROM workson INNER JOIN Employee ON workson.employee\_no = employee.employee\_no WHERE job = 'clerk' and department\_name = 'd1';

1. Get the names of projects on which two or more clerks are working.

**Ans** : SELECT COUNT() enterdates FROM workson INNER JOIN Employee ON workson.employee\_no = employee.employee\_no WHERE job = 'clerk' and department\_name = 'd1';

1. Get the first and last names of the employees that are manager and that work on project Mercury.

**Ans:** SELECT firstName,lastname FROM employee where employee\_no in (SELECT employee\_no from works\_on WHERE job = 'manager' and project\_no ='p2');

1. Get the first and last names of all employee who entered the project at the same time as at least one other employee.

**Ans**: SELECT DISTINCT firstName,lastname FROM employee where employee\_no in (SELECT employee\_no from works\_on WHERE COUNT(enter\_date)>=2);

1. Get the employee numbers of the employees living in the same location and belonging to the same department as one another.

**Ans**: SELECT emp\_no FROM workson LEFT OUTER JOIN Employee ON workson.dep\_name = employee.dp\_name WHERE workson.location = employee.location ;

1. Get the employee numbers of all employees belonging to the Marketing department.

Find two equivalent solutions using:

* 1. the JOIN operator
  2. The correlated subquery.

Modifying a Table’s Contents

1. Insert the data of a new employee called Julia Long, whose employee number is 1111. Her department number is not known yet.

Ans: INSERT INTO employee (1111, 'Julia', 'Long', NULL);

1. Create a new table called **emp\_d1\_d2** with all employees who work for department d1 or d1, and load the corresponding rows from the **employee** table.

Ans: CREATE TABLE emp\_d2\_d2 (emp\_no integer not null);

INSERT INTO emp\_d1\_d2 (emp\_no) SELECT emp\_no FROM Employee WHERE dept = d1 OR dept = d2;

1. Modify the job of all the employees in project p1 who are managers. They have to work as clerks from now on.

Ans: UPDATE Works\_on SET Job = 'Clerk' WHERE project\_no = 'P1';

1. The budgets of all projects are no longer determined. Assign all budgets the NULL value.

Ans: UPDATE Project SET Budget NULL;

1. Increase the budget of the project where the manager has the employee number 10102. The increase is 10%.

Ans :

|  |
| --- |
|  |
| UPDATE Project SET Budget = Budget + (Budget \* 0.1) WHERE project\_no IN (SELECT project\_no from Works\_on where emp\_no = 10102 and Job = 'manager'); |
|  |

1. Change the enter date for the projects for those employees who work in project p1 and belong to the department *Sales.* The new date is 12.12.1998.

Ans:

UPDATE Works\_on SET enter\_date = 1998-12-12 WHERE emp\_no in (SELECT emp\_no FROM Employee WHERE dept IN (SELECT dept from Departmnet where DeptType = 'sales' ));

1. Create a stored procedure to insert data into department and Employee table.

Ans:

CREATE PROCEDURE Insert IntoDepartment @DepId varchar(2), @DepName char(28), @Location char (40)

AS

INSERT INTO department (@DepId, @DepName, @Location);

GO;

(

CREATE PROCEDURE InsertIntoEmployee @EmpNo integer, @Emp\_fname char (20), @Emp\_Iname char(28), @DepId varchar(2)

AS

INSERT INTO Employee (@empNo, @emp\_Fnamne, @emp\_Iname, @DepId);

GO;