

# DBMS Assignment No 1.

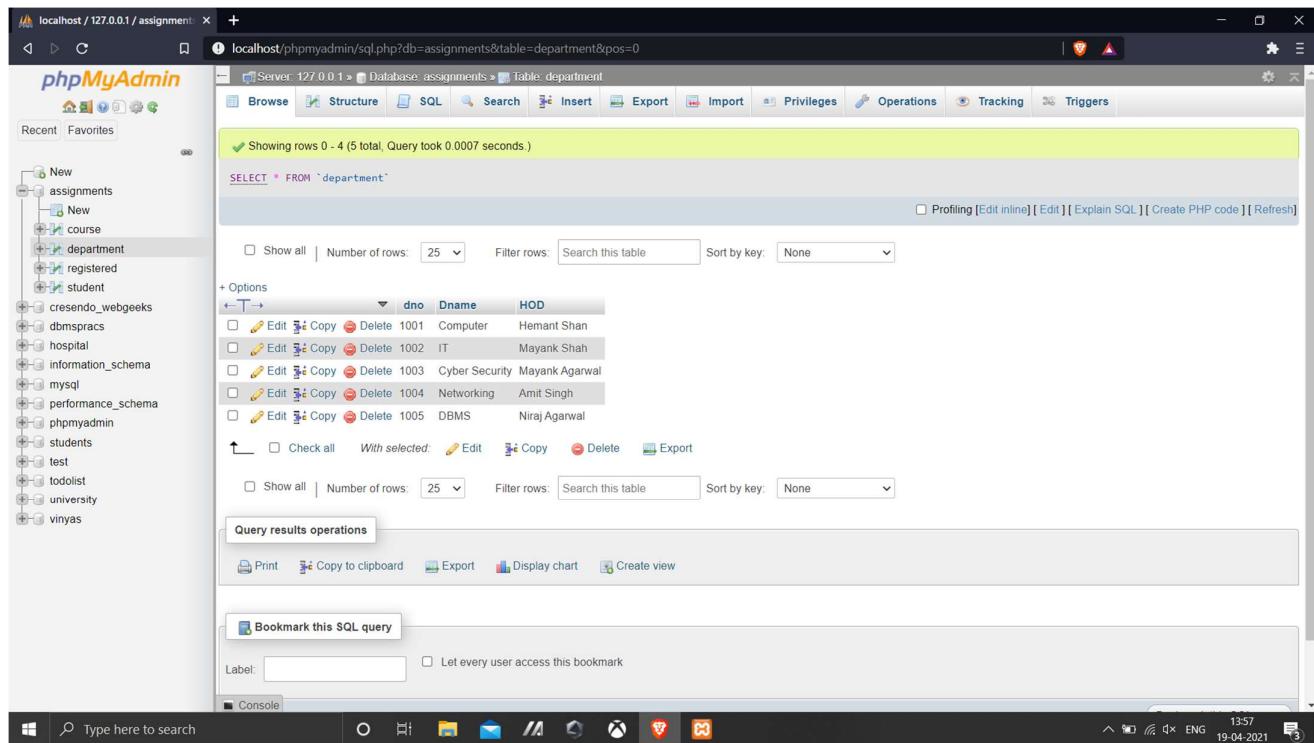
## Q1. Implement the following in MySQL:

A) Create the following Tables and Insert Values:

i) Table Department:

```
CREATE TABLE department(
    dno INT PRIMARY KEY NOT NULL,
    Dname VARCHAR(20) NOT NULL,
    HOD VARCHAR(20) NOT NULL
);
```

```
INSERT INTO `department` ('dno', 'Dname', 'HOD') VALUES
('1001','Computer','Hemant Shan'),
('1002','IT','Mayank Shah'),
('1003','Cyber Security','Mayank Agarwal'),
('1004','Networking','Amit Singh'),
('1005','DBMS','Niraj Agarwal');
```



The screenshot shows the phpMyAdmin interface for the 'assignments' database. The left sidebar lists various schemas and tables. The 'department' table is selected in the main area. A query result for the 'department' table is displayed, showing five rows of data. The columns are labeled 'dno', 'Dname', and 'HOD'. The data is as follows:

dno	Dname	HOD
1001	Computer	Hemant Shan
1002	IT	Mayank Shah
1003	Cyber Security	Mayank Agarwal
1004	Networking	Amit Singh
1005	DBMS	Niraj Agarwal

ii) Student Table:

CREATE TABLE student(

```
    reg_no VARCHAR(6) PRIMARY KEY NOT NULL,  
    name VARCHAR(20) NOT NULL,  
    age SMALLINT CHECK (age>0),  
    dob date NOT NULL,  
    cgpa decimal(3,2),  
    dno INT,  
    FOREIGN KEY (dno) REFERENCES department(dno)  
);
```

```
INSERT INTO `student` ('reg_no', 'name', 'age', 'dob', 'cgpa', 'dno') VALUES  
('st2101', 'Mansi Singh', '20', '2001-04-01', '8.5', '1002'),  
('st2102', 'Manish Jha', '18', '2003-02-05', '9.6', '1001'),  
('st2103', 'Yash Shah', '21', '2000-11-10', '9.2', '1003'),  
('st2104', 'Karuna Wagh', '19', '2001-06-15', '9.8', '1005'),  
('st2105', 'Ishaan Kishan', '25', '1997-04-25', '8.1', '1004'),  
('st2106', 'Charan Singh', '22', '1998-04-01', '4.2', '1002'),  
('st2107', 'Ivan Sayross', '26', '1995-05-04', '7.3', '1004');
```

The screenshot shows the phpMyAdmin interface for a MySQL database named 'assignments'. The left sidebar shows the database structure with a 'student' table selected. The main area displays the contents of the 'student' table. The table has columns: reg\_no, name, age, dob, cgpa, and dno. The data consists of seven rows, each representing a student with their respective details.

	reg_no	name	age	dob	cgpa	dno
1	st2101	Mansi Singh	20	2001-04-01	8.5	1002
2	st2102	Manish Jha	18	2003-02-05	9.6	1001
3	st2103	Yash Shah	21	2000-11-10	9.2	1003
4	st2104	Karuna Wagh	19	2001-06-15	9.8	1005
5	st2105	Ishaan Kishan	25	1997-04-25	8.1	1004
6	st2106	Charan Singh	22	1998-04-01	4.2	1002
7	st2107	Ivan Sayross	26	1995-05-04	7.3	1004

### iii) Courses Table:

CREATE TABLE course(

```
    course_code VARCHAR(6) PRIMARY KEY NOT NULL,  
    title VARCHAR(20) NOT NULL,  
    credit SMALLINT CHECK (credit>0),  
    type VARCHAR(2) DEFAULT "FT"  
);
```

```
INSERT INTO `course` ('code', 'title', 'credit', 'type') VALUES  
('SWE001', 'Python Coding', '15', 'FT'),  
('SWE002', 'Cyber Security', '15', 'PT'),  
('SWE003', 'SQL Syntax', '15', 'FT'),  
('SWE004', 'Networking', '15', 'PT'),  
('SWE005', 'Blockchain', '15', 'FT');
```

The screenshot shows the phpMyAdmin interface for a MySQL database named 'assignments'. The left sidebar shows various databases and tables, with 'course' selected. The main area displays the contents of the 'course' table. The table has four columns: code, title, credit, and type. The data is as follows:

	code	title	credit	type
1	SWE001	Python Coding	15	FT
2	SWE002	Cyber Security	15	PT
3	SWE003	SQL Syntax	15	FT
4	SWE004	Networking	15	PT
5	SWE005	Blockchain	15	FT

iv) Registered Table:

```
CREATE TABLE registered(
    code VARCHAR(6) NOT NULL,
    reg_no VARCHAR(6) NOT NULL,
    venue VARCHAR(6) NOT NULL,
    slots SMALLINT NOT NULL CHECK (slots>0),
    PRIMARY KEY (code,reg_no),
    FOREIGN KEY (code) REFERENCES course(code),
    FOREIGN KEY (reg_no) REFERENCES student(reg_no)
);
```

```
INSERT INTO `registered` ('code', 'reg_no', 'venue', 'slots') VALUES
('SWE001', 'st2101', 'lab201', '3'),
('SWE002', 'st2101', 'lab202', '2'),
('SWE003', 'st2102', 'lab101', '4'),
('SWE004', 'st2102', 'lab501', '2'),
('SWE005', 'st2103', 'lab302', '5'),
('SWE002', 'st2103', 'lab202', '2'),
('SWE001', 'st2104', 'lab201', '3'),
('SWE004', 'st2104', 'lab501', '2'),
('SWE003', 'st2105', 'lab101', '4'),
('SWE002', 'st2105', 'lab202', '2');
```

The screenshot shows the phpMyAdmin interface for the 'registered' table. The table structure is as follows:

	code	reg_no	venue	slots
<input type="checkbox"/>	SWE001	st2101	lab201	3
<input type="checkbox"/>	SWE002	st2101	lab202	2
<input type="checkbox"/>	SWE003	st2102	lab101	4
<input type="checkbox"/>	SWE004	st2102	lab501	2
<input type="checkbox"/>	SWE005	st2103	lab302	5
<input type="checkbox"/>	SWE002	st2103	lab202	2
<input type="checkbox"/>	SWE001	st2104	lab201	3
<input type="checkbox"/>	SWE004	st2104	lab501	2
<input type="checkbox"/>	SWE003	st2105	lab101	4
<input type="checkbox"/>	SWE002	st2105	lab202	2

## B) Execute Queries for the following:

i) List the student details who registered Course code 'SWE1001':

```
SELECT *
FROM student,registered
WHERE student.reg_no = registered.reg_no
AND registered.code = "SWE001";
```

The screenshot shows the phpMyAdmin interface on a Windows desktop. The left sidebar lists databases and tables under the 'assignments' database. The main area displays the results of the executed SQL query.

**Query Results:**

```
Showing rows 0 - 1 (2 total). Query took 0.0031 seconds.

SELECT * FROM student,registered WHERE student.reg_no = registered.reg_no AND registered.code = "SWE001"
```

**Table Data:**

reg_no	name	age	dob	cgpa	dno	code	reg_no	venue	slots
st2101	Mansi Singh	20	2001-04-01	8.50	1002	SWE001	st2101	lab201	3
st2104	Karuna Wagh	19	2001-06-15	9.80	1005	SWE001	st2104	lab201	3

**Operations:**

- Print
- Copy to clipboard
- Export
- Display chart
- Create view

**Bookmark this SQL query:**

Label:   Let every user access this bookmark

Console

Type here to search

Windows taskbar icons: File Explorer, Mail, Task View, Task Manager, Taskbar settings, Start button, Date/Time (14:29, 19-04-2021), Battery status.

ii) List the number of students in each course.:

```
SELECT code,COUNT(reg_no)
FROM registered
GROUP BY code;
```

The screenshot shows the phpMyAdmin interface on a Windows desktop. The left sidebar lists databases: New, assignments, dbmspracs, hospital, information\_schema, mysql, performance\_schema, phpmyadmin, students, test, todolist, university, and vinyas. The 'assignments' database is selected. In the center, the 'registered' table is shown. A query has been run:

```
SELECT code,COUNT(reg_no) FROM registered GROUP BY code
```

The results are displayed in a table:

code	COUNT(reg_no)
SWE001	2
SWE002	3
SWE003	2
SWE004	2
SWE005	1

Below the table are options to Print, Copy to clipboard, Export, Display chart, and Create view. There is also a 'Bookmark this SQL query' section with a 'Label:' input field and a checkbox for 'Let every user access this bookmark'. At the bottom right is a 'Bookmark this SQL query' button.

iii) List the department no which are having less than 2 students.:

```
SELECT dno,COUNT(reg_no)
FROM student
GROUP BY dno
HAVING COUNT(reg_no)<2;
```

The screenshot shows the phpMyAdmin interface on a Windows desktop. The left sidebar lists databases and tables, including 'assignments' and its 'student' table. The main area displays the results of the executed SQL query:

```
SELECT dno,COUNT(reg_no) FROM student GROUP BY dno HAVING COUNT(reg_no)<2
```

The results table shows three rows:

dno	COUNT(reg_no)
1001	1
1003	1
1005	1

Below the results, there are options for 'Query results operations' such as Print, Copy to clipboard, Export, Display chart, and Create view. A 'Bookmark this SQL query' section is also present.

iv) List the student name, department name, HOD and the courses registered by him/her who is having more than 9.0 CGPA:

SELECT

student.reg\_no,student.name,department.dno,department.Dname,department.HOD,registered.code

FROM student NATURAL JOIN department NATURAL JOIN registered

WHERE student.cgpa >=9.0 ;

The screenshot shows the phpMyAdmin interface on a Windows desktop. The left sidebar lists databases and tables, with 'student' selected. The main area displays a query results page for the executed SQL statement. The results table has columns: reg\_no, name, dno, Dname, HOD, and code. The data shows 6 rows of student information joined with their respective departments and registered courses.

reg_no	name	dno	Dname	HOD	code
st2102	Manish Jha	1001	Computer	Hemant Shan	SWE003
st2102	Manish Jha	1001	Computer	Hemant Shan	SWE004
st2103	Yash Shah	1003	Cyber Security	Mayank Agarwal	SWE002
st2103	Yash Shah	1003	Cyber Security	Mayank Agarwal	SWE005
st2104	Karuna Wagh	1005	DBMS	Niraj Agarwal	SWE001
st2104	Karuna Wagh	1005	DBMS	Niraj Agarwal	SWE004

v) Retrieve the average CGPA of each department:

```
SELECT dno, AVG(cgpa)
FROM student
GROUP BY dno;
```

The screenshot shows the phpMyAdmin interface on a Windows desktop. The left sidebar lists databases: assignments, dbmspracs, hospital, information\_schema, mysql, performance\_schema, phpmyadmin, students, test, todolist, university, and vinyas. The 'assignments' database is selected. The main area displays the 'student' table with the following data:

dno	AVG(cgpa)
1001	9.600000
1002	6.350000
1003	9.200000
1004	7.700000
1005	9.800000

Below the table, there are buttons for Print, Copy to clipboard, Export, Display chart, and Create view.

vi) Retrieve the student register numbers who are not registered for any course:

```
SELECT reg_no, name  
FROM student  
WHERE reg_no NOT IN (  
    SELECT reg_no  
    FROM registered;  
);
```

The screenshot shows the phpMyAdmin interface on a Windows desktop. The left sidebar lists databases and tables, with 'student' selected. The main area displays the results of the executed SQL query:

```
SELECT reg_no, name FROM student WHERE reg_no NOT IN (SELECT reg_no FROM registered)
```

The results table shows two rows:

reg_no	name
st2106	Charan Singh
st2107	Ivan Sayross

Below the table are various operations buttons: Edit, Copy, Delete, and Export.

## Q2. Implement the following in MySQL:

A) Create the following Tables and Insert Values:

i) Table Department:

CREATE TABLE department(

    dno INT PRIMARY KEY NOT NULL,  
    dname VARCHAR(20),  
    mgr\_id int

);

```
INSERT INTO `department` ('dno', 'dname', 'mgr_id') VALUES  
('1001', 'Marketing', '5001'),  
('1002', 'Sales', '5005'),  
('1003', 'Research', '5002'),  
('1004', 'Manufacturing', '5003'),  
('1005', 'Shipping', '5004');
```

ALTER TABLE department ADD FOREIGN KEY (mgr\_id) REFERENCES employee(id);

The screenshot shows the phpMyAdmin interface for the 'assignment' database. The left sidebar shows the database structure with tables like 'New', 'assignment', 'department', 'employee', 'project', 'works\_on', 'assignments', 'cresendo\_webgeeks', 'dbmspracs', 'hospital', 'information\_schema', 'mysql', 'performance\_schema', 'phpmyadmin', 'students', 'test', 'todolist', 'university', and 'vinyas'. The 'department' table is selected in the main area. The SQL tab displays the query: 'SELECT \* FROM `department`'. The results show five rows of data:

	dno	dname	mgr_id
1	1001	Marketing	5001
2	1002	Sales	5005
3	1003	Research	5002
4	1004	Manufacturing	5003
5	1005	Shipping	5004

Below the results, there are buttons for 'Edit', 'Copy', 'Delete', 'Check all', 'With selected:', and 'Export'. The bottom section includes 'Query results operations' with options for 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'. A 'Bookmark this SQL query' section allows labeling the query and setting access permissions.

## ii) Table Employee:

```
CREATE TABLE employee(
    id INT PRIMARY KEY NOT NULL,
    name VARCHAR(20),
    salary INT,
    dno INT,
    FOREIGN KEY (dno) REFERENCES department(dno)
);
```

```
INSERT INTO `employee` ('id', 'name', 'salary', 'dno') VALUES
('5001', 'Rahul Desai', '52500', '1001'),
('5002', 'Ivan Sayross', '68000', '1003'),
('5003', 'Yash Shah', '50500', '1004'),
('5004', 'Karuna Wagh', '61000', '1005'),
('5005', 'Janhavi Jadhav', '45000', '1002'),
('5006', 'Gopali Kamat', '75000', '1003'),
('5007', 'Rhutuja Vispute', '65000', '1001'),
('5008', 'Yashraj Sawant', '57000', '1002'),
('5009', 'Raman Saiva', '63500', '1004'),
('5010', 'Chinmay Kelkar', '54000', '1005');
```

The screenshot shows the phpMyAdmin interface for the 'assignment' database. The left sidebar lists databases and tables, with 'employee' selected. The main area displays the contents of the 'employee' table. The table has columns: id, name, salary, and dno. The data consists of 10 rows, each representing an employee with their ID, name, salary, and department ID.

	id	name	salary	dno
<input type="checkbox"/>	5001	Rahul Desai	52500	1001
<input type="checkbox"/>	5002	Ivan Sayross	68000	1003
<input type="checkbox"/>	5003	Yash Shah	50500	1004
<input type="checkbox"/>	5004	Karuna Wagh	61000	1005
<input type="checkbox"/>	5005	Janhavi Jadhav	45000	1002
<input type="checkbox"/>	5006	Gopali Kamat	75000	1003
<input type="checkbox"/>	5007	Rhutuja Vispute	65000	1001
<input type="checkbox"/>	5008	Yashraj Sawant	57000	1002
<input type="checkbox"/>	5009	Raman Saiva	63500	1004
<input type="checkbox"/>	5010	Chinmay Kelkar	54000	1005

### iii) Table Projects:

```
CREATE TABLE project(
    pno INT PRIMARY KEY NOT NULL,
    pname VARCHAR(20),
    dno INT,
    FOREIGN KEY (dno) REFERENCES department(dno)
);
```

```
INSERT INTO `project` ('pno', 'pname', 'dno') VALUES
('6001', 'Flamethrower', '1003'),
('6002', 'Silicon MP', '1003'),
('6003', 'OLED Display', '1003'),
('6004', 'Under Display Camera', '1003'),
('6005', 'Ad Campaign', '1001'),
('6006', 'Survey Feed Back', '1001'),
('6007', 'New Box Design', '1004'),
('6008', 'Eco friendly box', '1004');
```

The screenshot shows the phpMyAdmin interface for the 'assignment' database. The 'project' table is selected. The table structure includes columns: pno, pname, and dno. The data consists of 8 rows, each representing a project with its name and assigned department ID.

	pno	pname	dno
<input type="checkbox"/>	6001	Flamethrower	1003
<input type="checkbox"/>	6002	Silicon MP	1003
<input type="checkbox"/>	6003	OLED Display	1003
<input type="checkbox"/>	6004	Under Display Camera	1003
<input type="checkbox"/>	6005	Ad Campaign	1001
<input type="checkbox"/>	6006	Survey Feed Back	1001
<input type="checkbox"/>	6007	New Box Design	1004
<input type="checkbox"/>	6008	Eco friendly box	1004

i) Table Works\_on:

```
CREATE TABLE works_on(
    emp_id INT NOT NULL,
    pno INT NOT NULL,
    no_of_hours INT NOT NULL,
    FOREIGN KEY (pno) REFERENCES project(pno),
    FOREIGN KEY (emp_id) REFERENCES employee(id),
    PRIMARY KEY (emp_id,pno)
);
```

```
INSERT INTO `works_on` ('emp_id', 'pno', 'no_of_hours') VALUES
('5006', '6002', '40'),
('5006', '6004', '80'),
('5006', '6003', '80'),
('5002', '6001', '50'),
('5007', '6005', '60'),
('5001', '6006', '30'),
('5009', '6008', '50'),
('5009', '6007', '50');
```

The screenshot shows the phpMyAdmin interface for the 'assignment' database. The left sidebar lists databases and tables, with 'works\_on' selected. The main area displays the contents of the 'works\_on' table. A green status bar at the top indicates 'Showing rows 0 - 7 (8 total, Query took 0.0016 seconds.)'. Below it, a SQL query 'SELECT \* FROM `works\_on`' is shown. The table data is presented in a grid with columns: emp\_id, pno, and no\_of\_hours. Each row has edit, copy, and delete options. The data matches the INSERT statements provided in the text above.

emp_id	pno	no_of_hours
5001	6006	30
5002	6001	50
5006	6002	40
5006	6003	80
5006	6004	80
5007	6005	60
5009	6006	50
5009	6008	50

## B) Execute Queries for the following:

- i) List the employee id who is working on more than 3 projects.:

```
SELECT emp_id,COUNT(pno)
FROM works_on
GROUP BY emp_id
HAVING COUNT(pno)>=3;
```

The screenshot shows the phpMyAdmin interface on a Windows desktop. The left sidebar lists databases and tables, with 'assignment' selected. The main area displays the results of the executed SQL query:

```
SELECT emp_id,COUNT(pno) FROM works_on GROUP BY emp_id HAVING COUNT(pno)>=3
```

emp_id	COUNT(pno)
5006	3

Below the table are 'Query results operations' buttons: Print, Copy to clipboard, Export, Display chart, Create view. A 'Bookmark this SQL query' section is also present.

The taskbar at the bottom shows various pinned icons, and the system tray indicates the date as 19-04-2021 and time as 16:31.

ii) Retrieve the number of projects in each department.

```
SELECT dno,COUNT(pno) AS number_of_projects  
FROM project  
GROUP BY dno;
```

The screenshot shows the phpMyAdmin interface on a Windows desktop. The left sidebar lists databases: New, assignment, assignments, dbmspracs, hospital, information\_schema, mysql, performance\_schema, phpmyadmin, students, test, todolist, university, and vinyas. The 'assignment' database is selected. In the center, the 'project' table is selected under the 'assignment' database. The 'Browse' tab is active, displaying the results of the query: 'SELECT dno,COUNT(pno) AS number\_of\_projects FROM project GROUP BY dno'. The results table shows three rows:

dno	number_of_projects
1001	2
1003	4
1004	2

Below the table are 'Query results operations' buttons: Print, Copy to clipboard, Export, Display chart, and Create view. At the bottom, there is a 'Bookmark this SQL query' section with a 'Label:' input field and a checkbox 'Let every user access this bookmark'.

iii) Retrieve the employee id, name, salary, dname, manager id and the project name he/she has been working on:

```
SELECT
employee.id,employee.name,employee.salary,department.dname,department.mgr_id,project.pname
FROM employee,department,works_on,project
WHERE employee.dno = department.dno and employee.id = works_on.emp_id and
works_on.pno = project.pno
```

The screenshot shows the phpMyAdmin interface with the following details:

- Left Panel:** Shows the database structure with the 'assignment' schema selected. It contains tables: New, department, employee, project, and works\_on.
- Top Bar:** Shows tabs for 'localhost / 127.0.0.1 / assignment' and 'localhost / 127.0.0.1 / assignment / en'. The URL in the address bar is 'localhost/phpmyadmin/db\_sql.php?db=assignment'.
- Main Content:**
  - A green message box at the top says 'Showing rows 0 - 7 (8 total). Query took 0.0087 seconds.'
  - The SQL query entered is: 'SELECT employee.id,employee.name,employee.salary,department.dname,department.mgr\_id,project.pname FROM employee,department,works\_on,project WHERE employee.dno = department.dno and employee.id = works\_on.emp\_id and works\_on.pno = project.pno'
  - The results table displays 8 rows of data:

id	name	salary	dname	mgr_id	pname
5001	Rahul Desai	52500	Marketing	5001	Survey Feed Back
5007	Rhutuja Vispute	65000	Marketing	5001	Ad Campaign
5002	Ivan Sayross	68000	Research	5002	Flamethrower
5006	Gopali Kamat	75000	Research	5002	Silicon MP
5006	Gopali Kamat	75000	Research	5002	OLED Display
5009	Gopali Kamat	75000	Research	5002	Under Display Camera
5009	Raman Saiva	63500	Manufacturing	5003	New Box Design
5009	Raman Saiva	63500	Manufacturing	5003	Eco friendly box

  - Bottom Buttons:** Includes 'Query results operations' (Print, Copy to clipboard, Export, Display chart, Create view), 'Bookmark this SQL query', and a 'Console' tab.

- iv) Retrieve the employee details who are working on Department 'Marketing' and getting the salary more than 50000:

```
SELECT employee.id,employee.name,employee.salary,department.dname  
FROM employee,department  
WHERE employee.dno = department.dno AND employee.salary >=50000 AND  
department.dname = "Marketing"
```

The screenshot shows the phpMyAdmin interface on a Windows desktop. The database selected is 'assignment'. The left sidebar lists various databases and tables. The main area displays the results of a SQL query:

```
SELECT employee.id,employee.name,employee.salary,department.dname FROM employee,department WHERE employee.dno = department.dno AND employee.salary >=50000 AND department.dname = "Marketing".
```

The results show two rows:

id	name	salary	dname
5001	Rahul Desai	52500	Marketing
5007	Rhutuja Vispute	65000	Marketing

Below the table are options to print, copy to clipboard, export, display chart, or create a view. There is also a 'Bookmark this SQL query' button.

v) Retrieve the department numbers which does not have any project.:

```
SELECT *
FROM department
WHERE department.dno NOT IN
(
    SELECT project.dno FROM project
);
```

The screenshot shows the phpMyAdmin interface on a Windows desktop. The left sidebar lists databases and tables, with 'assignment' selected. The main area displays a query results page for the 'department' table. The query executed was:

```
SELECT * FROM department WHERE department.dno NOT IN ( SELECT project.dno FROM project )
```

The results show two rows from the 'department' table:

dno	dname	mgr_id
1002	Sales	5005
1005	Shipping	5004

Below the table, there are options for 'Query results operations' including Print, Copy to clipboard, Export, Display chart, and Create view. A 'Bookmark this SQL query' section is also present.

vi) Retrieve all the department numbers, department name and controlling project number, project name if any:

```
SELECT department.dno, department.dname, project.pno, project.pname  
FROM department ,project  
WHERE department.dno = project.dno;
```

The screenshot shows the phpMyAdmin interface with the following details:

- Server:** 127.0.0.1
- Database:** assignment
- Table:** department
- Query Results:**
  - Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.
  - Showing rows 0 - 7 (8 total). Query took 0.0048 seconds.
  - SQL Query:

```
SELECT department.dno,department.dname,project.pno,project.pname FROM department ,project WHERE department.dno = project.dno
```
  - Operations:
    - Show all
    - Number of rows: 25
    - Filter rows: Search this table
  - Query results operations:
    - Print
    - Copy to clipboard
    - Export
    - Display chart
    - Create view
- Console:** mark this SQL query

Q3

- (1) List the student details who registered in course code = 'SWE1001'.

Ans

$\Pi_{x} \{ \sigma_{r \text{ code} = 'SWE1001'} (P_s(\text{student}) \bowtie P_r(\text{Registered})) \}$

- (2) List the number of students in each course

Ans

$\text{code} \{ \text{Count}(\text{regno}) (\text{Registered}) \}$

OR.

$\text{code}, \text{count}(\text{Regno}) (\text{Registered})$

- (3) List the department no which are having less than 2 students.

$\text{dno}, \text{count}(\text{Regno}) \leq 2 (\text{Registered})$

OR

$\text{dno} \{ \text{Count}(\text{Regno}) \leq 2 (\text{Registered}) \}$

- (4) List the Student name, department name, HOD, and course registered by student having more than 9.0 CGPA

$\{ \text{S-name}, \text{d-name}, \text{d-HOD}, \text{r-code} \}$

$\sigma_{\text{CGPA} = 9.0} ( \{ \text{d-name} \text{ (department)} \bowtie \{ \text{S-name} \text{ (student)} \bowtie \{ \text{r-code} \text{ (registered)} \} \} )$

(15) Retrieve the average CGPA of each department  
Ans

$\text{Y}(\text{dno}, \text{avg}(\text{CGPA})) \text{ (Student)}$

OR

$\text{dno} (\text{G}(\text{avg}(\text{CGPA}))) \text{ (Student)}$

(16) Retrieve the student register number who are not registered for any course

Ans

$[\text{IT.Regno} \text{ (Student)}] - [\text{IT.Regno} \text{ (Registered)}]$

Q4

- (1) List the employee id who is working on more than 3 projects

Ans

$\{ \text{empid} \mid \text{count}(\text{Pro}) >= 3 \text{ (Works\_on)} \}$

OR.

$\text{empid} \mid \text{count}(\text{Pro}) >= 3 \text{ (works\_on)}.$

- (2) Retrieve the number of projects in each department.

Ans

$\{ \text{dno}, \text{count}(\text{Pro}) \mid (\text{Project}) \}$

OR

$\text{dno} \mid \text{count}(\text{Pro}) \mid (\text{Project})$

- (3) Retrieve the employee id, name, salary, d-name, manager id and project name working on.

Ans

$\{ \text{e.id}, \text{e.name}, \text{e.salary}, \text{d.dname}, \text{d.mgr_id}, \text{p.pname} \mid$

$P_e(\text{employee}) \bowtie P_w(\text{works\_on}) \bowtie P_p(\text{Project})$   
 $e.id = w.empid \quad w.pro = p.pno$

$\bowtie p.dno = d.dno \text{ (department.) } \}$

4 Retrieve the employee details whose working on Department 'Marketing' and getting salary more than 50000.

Ans

$$\Pi * (\sigma_{d.dname = 'Marketing'} \wedge s.salary > 50000)$$
$$P_d(\text{department}) \bowtie P_s(\text{salary})$$

5 Retrieve the department numbers which does not have any project.

Ans

$$[\Pi_{dno}(\text{Department})] - [\Pi_{dno}(\text{Project})]$$

(6) Retrieve all the department numbers, department names and controlling project numbers, project name if any.

Ans

$$\Pi_{dno, d.dname, p.pno, p.pname}$$
$$P_p(\text{Project}) \bowtie P_d(\text{Department})$$