



# **Discipline**

## **Queries**

*(3 versions)*

## **CS5201**

## **Stage-2**

**Sheikh Muhammed Tadeeb (AU19B1014)**

## ❖ Queries Version-1:

### ➤ Queries:

1. How shall a new user register to the application.
2. How shall a user recover his account if he forgets his password.
3. How shall we authenticate that user is valid or not.
4. How shall a user maintain entry of goods that are purchased & sold by him.
5. How shall a user review the record of goods he/she has purchased or sold.

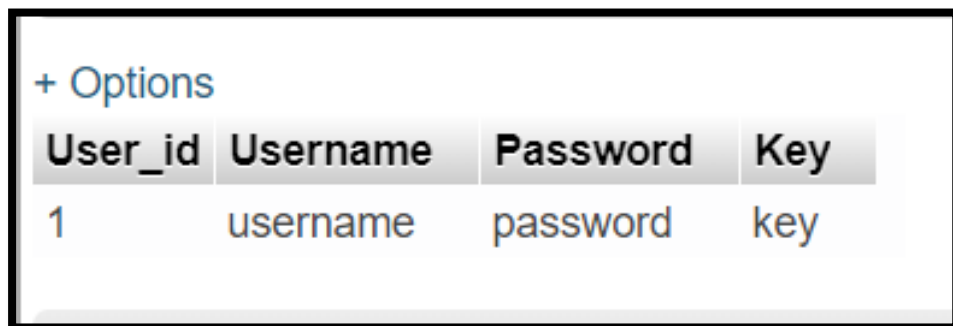
### ➤ Query Description:

- The places where we require user input must be attached with “insert” command in database.
- The places where we need authentication must be attached with “Select” command along with “where” clause which matches the specified conditions.

### ➤ SQL queries:

1. Insert into USER values (user\_id, 'username', 'password', 'key');

#### Output:



The screenshot shows a database table with four columns: User\_id, Username, Password, and Key. The first row contains the values 1, username, password, and key. Above the table, there is a link labeled '+ Options'.

User_id	Username	Password	Key
1	username	password	key

2. Select user\_id from USER where username = 'enteredValue' AND password = 'enteredValue';

### Output:

✓ Showing rows 0 - 0 (1 total, Query took 0.0029 seconds.)

`Select User_id from USER where Username = 'username' AND Password= 'password';`

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows: 25  Filter rows:

+ Options

User_id
1

3. Select user\_id from USER where Reckey = 'enteredKey';

### Output:

✓ Showing rows 0 - 0 (1 total, Query took 0.0025 seconds.)

`Select user_id from USER where Reckey = 'key';`

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows: 25  Filter rows:

+ Options

user_id
1

4. Insert into GOODS values (goods\_id, 'goods\_name', 'goods\_type');

### Output:

✓ 1 row inserted. (Query took 0.0304 seconds.)

`Insert into GOODS(Goods_name,Goods_type) values ('Screw', 'Mettalic');`

[\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Create PHP code \]](#)

✓ Showing rows 0 - 0 (1 total, Query took 0.0008 seconds.)

`SELECT * from goods;`

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows: 25  Filter rows:

+ Options

	Goods_id	Goods_type	Goods_name
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	11	Mettalic	Screw

5. Insert into PURCHASED (Goods\_id, P\_qty, P\_date, P\_time, P\_cost) values (goods\_id, P\_qty, 'P\_date', 'P\_time', P\_cost);

### Output:

✓ 1 row inserted. (Query took 0.1134 seconds.)

```
Insert into PURCHASED (Goods_id, P_qty, P_date, P_time, P_cost) values (11, 20, '2000-02-11', '11:30:20', 300);
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ Showing rows 0 - 0 (1 total, Query took 0.0032 seconds.)

```
SELECT * FROM `purchased`;
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

	Goods_id	P_qty	P_date	P_time	P_cost
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	11	20	2000-02-11	11:30:20.00000	300

☐ Check all | With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

☐ Show all | Number of rows: 25 | Filter rows: Search this table

6. Insert into SOLD (Goods\_id, S\_qty, S\_date, S\_time, S\_cost) values (goods\_id, S\_qty, 'S\_date', 'S\_time', S\_cost);

### Output:

✓ 1 row inserted. (Query took 0.0322 seconds.)

```
Insert into Sold (Goods_id, S_qty, S_date, S_time, S_cost) values (11, 20, '2000-03-12', '01:00:00', 10);
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 0 (1 total, Query took 0.0020 seconds.)

```
SELECT * FROM `sold`;
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

Goods_id	S_qty	S_date	S_time	S_cost
11	20	2000-03-12	01:00:00.000000	10

7. Select \* from PURCHASED Inner join goods ON goods.goods\_id = purchased.goods\_id where goods\_name = 'enteredName';

### Output:

✓ Showing rows 0 - 0 (1 total, Query took 0.0025 seconds.)

```
SELECT * FROM purchased INNER JOIN goods ON purchased.Goods_id = goods.Goods_id where goods_name = 'Screw';
```

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows: 25 ▾ Filter rows:

+ Options

Goods_id	P_qty	P_date	P_time	P_cost	Goods_id	Goods_type	Goods_name
11	20	2000-02-11	11:30:20.00000	300	11	Mettalic	Screw

☐ Show all | Number of rows: 25 ▾ Filter rows:

8. Select \* from SOLD Inner join goods ON goods.goods\_id = sold.goods\_id where goods\_name = 'enteredName';

### Output:

✓ Showing rows 0 - 0 (1 total, Query took 0.0024 seconds.)

```
SELECT * FROM Sold INNER JOIN goods ON sold.Goods_id = goods.Goods_id where goods_name = 'Screw';
```

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows: 25 ▾ Filter rows:

+ Options

Goods_id	S_qty	S_date	S_time	S_cost	Goods_id	Goods_type	Goods_name
11	20	2000-03-12	01:00:00.000000	10	11	Mettalic	Screw

## ❖ Queries Version-2:

### ➤ Queries:

1. How shall a user detect the credit he/she have on his/her business.
2. How shall a user detect the credit duration on his/her business.
3. How shall a user detect the credit he/she have on his/her customers.
4. How shall a user detect the credit duration on his/her customers.

### ➤ Query Description:

- A SQL Join statement is used to combine data or rows from two or more tables based on a common field between them. Different types of Joins are:
  1. INNER JOIN: The “inner join” keyword selects all rows from both the tables as long as the condition satisfies. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e. value of the common field will be same.
  2. LEFT JOIN: This join returns all the rows of the table on the left side of the join and matching rows for the table on the right side of join. The rows for which there is no matching row on right side, the result-set will contain null. “LEFT JOIN” is also known as “LEFT OUTER JOIN”.
  3. RIGHT JOIN: Right join is similar to Left join. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of join. The rows for which there is no matching row on left side, the result-set will contain null. RIGHT JOIN is also known as RIGHT OUTER JOIN.
  4. FULL JOIN: Full join creates the result-set by combining result of both “left join and right join”. The result-set will contain all the rows from both the

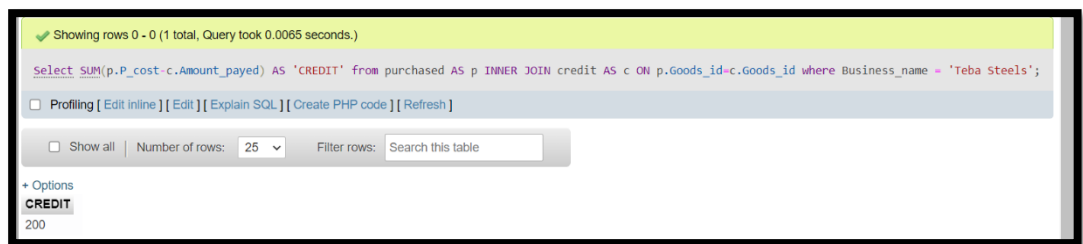
tables. The rows for which there is no matching, the result-set will contain NULL values.

- The places where we need get the data we must use “Select” command along with “where” clause which matches the specified conditions.

### ➤ SQL queries:

1. Select SUM (P. P\_cost – C.Amount\_paid) AS ‘CREDIT’ from PURCHASED AS P INNER JOIN CREDIT AS C ON P.Goods\_id = C.Goods\_id where Business\_name = ‘enteredName’;

#### Output:

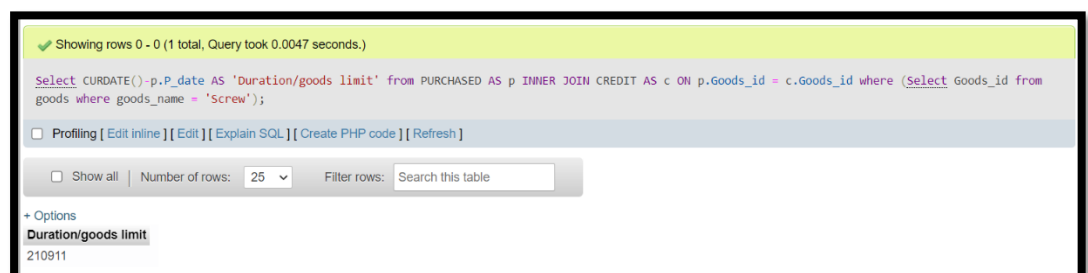


The screenshot shows a SQL query execution interface. At the top, it says "Showing rows 0 - 0 (1 total, Query took 0.0065 seconds.)". Below this is the SQL query: `Select SUM(p.P_cost-c.Amount_paid) AS 'CREDIT' from purchased AS p INNER JOIN credit AS c ON p.Goods_id=c.Goods_id where Business_name = 'Teba Steels';`. There are links for "Profiling", "Edit inline", "Edit", "Explain SQL", "Create PHP code", and "Refresh". Below the query is a control bar with "Show all", "Number of rows: 25", and "Filter rows: Search this table". Underneath, there is a section for "Options" with a table header "CREDIT" and a single row with the value "200".

CREDIT
200

2. Select CURDATE()-p.P\_date AS 'Duration/goods limit in days' , CURTIME()-p.P\_time AS 'Duration/goods limit in hours' from PURCHASED AS p INNER JOIN CREDIT AS c ON p.Goods\_id = c.Goods\_id where (Select Goods\_id from goods where goods\_name = 'enteredName');

#### Output:

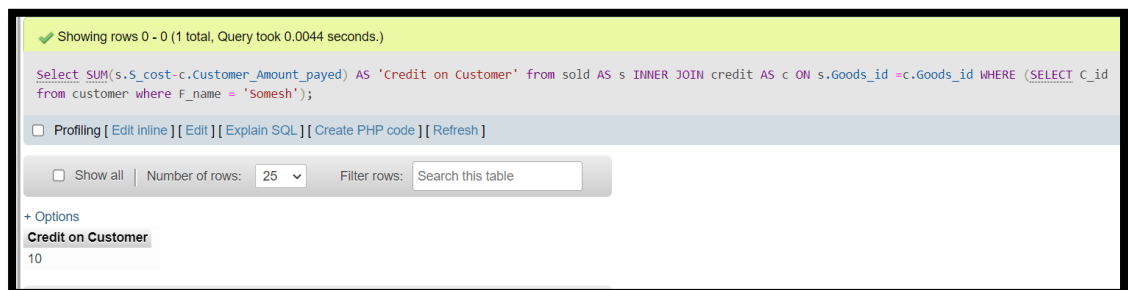


The screenshot shows a SQL query execution interface. At the top, it says "Showing rows 0 - 0 (1 total, Query took 0.0047 seconds.)". Below this is the SQL query: `Select CURDATE()-p.P_date AS 'Duration/goods limit' from PURCHASED AS p INNER JOIN CREDIT AS c ON p.Goods_id = c.Goods_id where (Select Goods_id from goods where goods_name = 'Screw');`. There are links for "Profiling", "Edit inline", "Edit", "Explain SQL", "Create PHP code", and "Refresh". Below the query is a control bar with "Show all", "Number of rows: 25", and "Filter rows: Search this table". Underneath, there is a section for "Options" with a table header "Duration/goods limit" and a single row with the value "210911".

Duration/goods limit
210911

3. Select SUM(s.S\_cost - c.Customer\_Amount\_paid) AS 'Credit on Customer' from sold AS s INNER JOIN credit AS c ON s.Goods\_id = c.Goods\_id where (Select C\_id from customer where F\_name = 'enteredName');

### Output:



Showing rows 0 - 0 (1 total, Query took 0.0044 seconds.)

```
Select SUM(s.S_cost-c.Customer_Amount_paid) AS 'Credit on Customer' from sold AS s INNER JOIN credit AS c ON s.Goods_id =c.Goods_id WHERE (SELECT C_id from customer where F_name = 'Somesh');
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

Credit on Customer
10

4. SELECT Curdate()-s.S\_date AS 'Duaration/Credit limit in days', CURTIME()-s.S\_time AS 'Duaration/Credit limit in Hours ' from sold AS s INNER JOIN credit AS c ON s.Goods\_id=c.Goods\_id where (Select C\_id from customer where F\_name= 'Somesh');

### Output:



Showing rows 0 - 0 (1 total, Query took 0.0229 seconds.)

```
SELECT Curdate()-s.S_date AS 'Duaration/Credit limit in days', CURTIME()-s.S_time AS 'Duaration/Credit limit in Hours' from sold AS s INNER JOIN credit AS c ON s.Goods_id=c.Goods_id where (select c_id from customer where F_name= 'Somesh');
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

Duaration/Credit limit in days	Duaration/Credit limit in Hours
210810	225649.000000



## ❖ Queries Version-3:

### ➤ Queries:

1. How shall a user detect the profit he/she made through his/her business.
2. How shall a user detect the static duration of product in his/her shop.

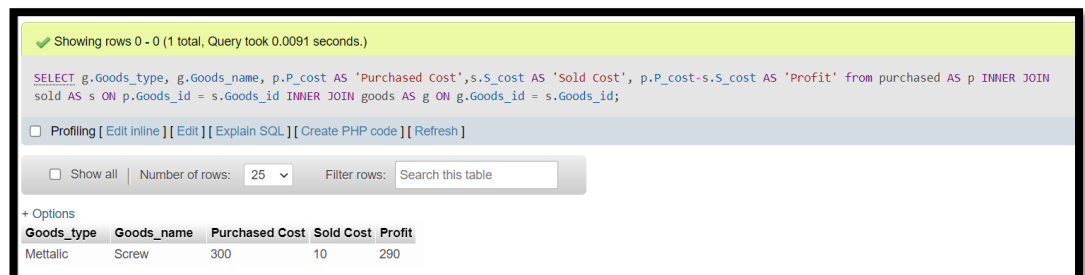
### ➤ Query Description:

- A SQL Join statement is used to combine data or rows from two or more tables based on a common field between them. *I have used joins on 3 tables here:*

### ➤ SQL queries:

1. `SELECT g.Goods_type, g.Goods_name, p.P_cost AS 'Purchased Cost', s.S_cost AS 'Sold Cost', p.P_cost - s.S_cost AS 'Profit' from purchased AS p INNER JOIN sold AS s ON p.Goods_id = s.Goods_id INNER JOIN goods AS g ON g.Goods_id = s.Goods_id;`

### Output:



Showing rows 0 - 0 (1 total, Query took 0.0091 seconds.)

```
SELECT g.Goods_type, g.Goods_name, p.P_cost AS 'Purchased Cost', s.S_cost AS 'Sold Cost', p.P_cost - s.S_cost AS 'Profit' from purchased AS p INNER JOIN sold AS s ON p.Goods_id = s.Goods_id INNER JOIN goods AS g ON g.Goods_id = s.Goods_id;
```

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

Goods_type	Goods_name	Purchased Cost	Sold Cost	Profit
Metallic	Screw	300	10	290

2. `SELECT g.Goods_type, g.Goods_name, p.P_date AS 'Purchased Date', s.S_date AS 'Sold Date', p.P_date - s.S_date AS 'Static limit (days)', s.S_date - p.P_date AS 'Static limit (Hrs)' from purchased AS p INNER JOIN sold AS s ON p.Goods_id = s.Goods_id INNER JOIN goods AS g ON g.Goods_id = s.Goods_id;`



Showing rows 0 - 0 (1 total, Query took 0.0093 seconds.)

```
SELECT g.Goods_type, g.Goods_name, p.P_date AS 'Purchased date', s.S_date AS 'Sold date', s.S_date - p.P_date AS 'Static Limit (Days)', s.S_time - p.P_time AS 'Static limit (hrs)' from purchased AS p INNER JOIN sold AS s ON p.Goods_id = s.Goods_id INNER JOIN goods AS g ON g.Goods_id = s.Goods_id;
```

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

Goods_type	Goods_name	Purchased date	Sold date	Static Limit (Days)	Static limit (hrs)
Metallic	Screw	2000-02-11	2000-03-12	101	-103020.000000