

ASSIGNMENT 4

GROUP 1

1. Order by Clause

1.1 Ascending Order

Query 1

```
1 use myHospital;
2 /*
3 1. ORDER BY CLAUSE
4 */
5 /* We are arranging whole Bills table in ascending and descending order on the basis of Date_Admitted */
6 select * from Bills order by Date_Admitted;
```

Result Grid

Bill_Id	Date_Admitted	Date_Discharged	Consultation_Fee	Pat_Id
807	2020-05-15	2020-05-30	11000	307
802	2020-05-31	2020-06-13	10500	302
803	2020-06-01	2020-06-06	9000	303
804	2020-06-03	2020-06-05	5000	304
801	2020-06-10	2020-06-19	10000	301
805	2020-06-10	2020-06-29	15000	305
806	2020-06-11	2020-06-15	8000	306
808	2020-06-11	2020-06-26	12500	308
809	2020-07-02	2020-07-04	4000	309
810	2020-07-03	2020-07-23	13000	310

Bills 1 x

Output

#	Time	Action	Message
1	18:20:49	use myHospital	0 row(s) affected
2	18:20:49	select * from Bills order by Date_Admitted LIMIT 0, 1000	10 row(s) returned

1.2 Descending Order

Query 1

```
1 use myHospital;
2 /*
3 1. ORDER BY CLAUSE
4 */
5 /* We are arranging whole Bills table in ascending and descending order on the basis of Date_Admitted */
6 select * from Bills order by Date_Admitted;
7 select * from Bills order by Date_Admitted desc;
```

Result Grid

Bill_Id	Date_Admitted	Date_Discharged	Consultation_Fee	Pat_Id
810	2020-07-03	2020-07-23	13000	310
809	2020-07-02	2020-07-04	4000	309
806	2020-06-11	2020-06-15	8000	306
808	2020-06-11	2020-06-26	12500	308
801	2020-06-10	2020-06-19	10000	301
805	2020-06-10	2020-06-29	15000	305
804	2020-06-03	2020-06-05	5000	304
803	2020-06-01	2020-06-06	9000	303
802	2020-05-31	2020-06-13	10500	302
807	2020-05-15	2020-05-30	11000	307

Bills 2 x

Output

#	Time	Action	Message
1	18:20:49	use myHospital	0 row(s) affected
2	18:20:49	select * from Bills order by Date_Admitted LIMIT 0, 1000	10 row(s) returned
3	18:22:33	select * from Bills order by Date_Admitted desc LIMIT 0, 1000	10 row(s) returned

2. Group by and Having

2.1 Group by

The screenshot shows a database query editor with a SQL query and its results. The query is as follows:

```
3 /* We are arranging whole Bills table in ascending and descending order on the basis of Date_Admitted */
4 * select * from Bills order by Date_Admitted;
5 * select * from Bills order by Date_Admitted desc;
6 /* 2. GROUP BY and HAVING */
7 /* We are grouping the count of Pat_Id based on the Sex of the Patient */
8 * select count(Pat_Id) from Patients group by Pat_Sex;
```

The results are displayed in a table with the following data:

count(Pat_Id)
5
5

The output pane shows the following messages:

#	Time	Action	Message
1	18:20:49	use myHospital	0 row(s) affected
2	18:20:49	select * from Bills order by Date_Admitted LIMIT 0, 1000	10 row(s) returned
3	18:22:33	select * from Bills order by Date_Admitted desc LIMIT 0, 1000	10 row(s) returned
4	18:29:19	select count(Pat_Id) from Pat_Id group by Pat_Sex LIMIT 0, 1000	Error Code: 1146. Table 'myhospital.pat_id' doesn't exist
5	18:29:50	select count(Pat_Id) from Patients group by Pat_Sex LIMIT 0, 1000	2 row(s) returned

2.2 Having

The screenshot shows a database query editor with a SQL query and its results. The query is as follows:

```
6 /* 2. GROUP BY and HAVING */
7 /* We are grouping the count of Pat_Id based on the Sex of the Patient */
8 * select count(Pat_Id) from Patients group by Pat_Sex;
9 /* We are grouping the count of Doc_Id based on sex of Doc and displaying number of Doc_Id which are greater than 5 */
10 /* As there are only 2 kinds of sex and there are 10 male and female doctors it will show 10 for both */
11 * select count(Doc_Id) from Doctors group by Doc_Sex having count(Doc_Id)>5;
```

The results are displayed in a table with the following data:

count(Doc_Id)
10
10

The output pane shows the following messages:

#	Time	Action	Message
1	18:20:49	use myHospital	0 row(s) affected
2	18:20:49	select * from Bills order by Date_Admitted LIMIT 0, 1000	10 row(s) returned
3	18:22:33	select * from Bills order by Date_Admitted desc LIMIT 0, 1000	10 row(s) returned
4	18:29:19	select count(Pat_Id) from Pat_Id group by Pat_Sex LIMIT 0, 1000	Error Code: 1146. Table 'myhospital.pat_id' doesn't exist
5	18:29:50	select count(Pat_Id) from Patients group by Pat_Sex LIMIT 0, 1000	2 row(s) returned
6	18:38:32	select count(Doc_Id) from Doctors group by Doc_Sex having count(Doc_Id)>5 LIMIT 0, 1000	2 row(s) returned

3. Aggregate Functions

3.1 Count Function

The screenshot shows a database query editor with a SQL query and its results. The query is as follows:

```
9 /* We are grouping the count of Doc_Id based on sex of Doc and displaying number of Doc_Id which are greater than 5 */
10 /* As there are only 2 kinds of sex and there are 10 male and female doctors it will show 10 for both */
11 select count(Doc_Id) from Doctors group by Doc_Sex having count(Doc_Id)>5;
12 /* 3. AGGREGATE FUNCTIONS , There are 5 aggregate functions, we will show all of them */
13 /* 3.1 COUNT FUNCTION , We will count the number of Pat_Age and that should give 10 as there are 10 patients in the Table */
14 select count(Pat_Age) from Patients;
```

The results grid shows the following data:

count(Pat_Age)
10

The output pane shows the following actions and messages:

#	Time	Action	Message
2	18:20:49	select * from Bills order by Date_Admitted LIMIT 0, 1000	10 row(s) returned
3	18:22:33	select * from Bills order by Date_Admitted desc LIMIT 0, 1000	10 row(s) returned
4	18:29:19	select count(Pat_Id) from Pat_Id group by Pat_Sex LIMIT 0, 1000	Error Code: 1146. Table 'myhospital.pat_id' doesn't exist
5	18:29:50	select count(Pat_Id) from Patients group by Pat_Sex LIMIT 0, 1000	2 row(s) returned
6	18:38:32	select count(Doc_Id) from Doctors group by Doc_Sex having count(Doc_Id)>5 LIMIT 0, 1000	2 row(s) returned
7	18:43:54	select count(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned

3.2 Sum Function

The screenshot shows a database query editor with a SQL query and its results. The query is as follows:

```
11 select count(Doc_Id) from Doctors group by Doc_Sex having count(Doc_Id)>5;
12 /* 3. AGGREGATE FUNCTIONS , There are 5 aggregate functions, we will show all of them */
13 /* 3.1 COUNT FUNCTION , We will count the number of Pat_Age and that should give 10 as there are 10 patients in the Table */
14 select count(Pat_Age) from Patients;
15 /* 3.2 SUM FUNCTION , We will calculate the sum of Pat_Age and that should give the sum of all 10 patients */
16 select sum(Pat_Age) from Patients;
```

The results grid shows the following data:

sum(Pat_Age)
501

The output pane shows the following actions and messages:

#	Time	Action	Message
3	18:22:33	select * from Bills order by Date_Admitted desc LIMIT 0, 1000	10 row(s) returned
4	18:29:19	select count(Pat_Id) from Pat_Id group by Pat_Sex LIMIT 0, 1000	Error Code: 1146. Table 'myhospital.pat_id' doesn't exist
5	18:29:50	select count(Pat_Id) from Patients group by Pat_Sex LIMIT 0, 1000	2 row(s) returned
6	18:38:32	select count(Doc_Id) from Doctors group by Doc_Sex having count(Doc_Id)>5 LIMIT 0, 1000	2 row(s) returned
7	18:43:54	select count(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
8	18:46:59	select sum(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned

3.3 AVG Function

The screenshot shows a query editor with a SQL query and its results. The query is as follows:

```
13 /* 3.1 COUNT FUNCTION , We will count the number of Pat_Age and that should give 10 as there are 10 patients in the Table */
14 • select count(Pat_Age) from Patients;
15 /* 3.2 SUM FUNCTION , We will calculate the sum of Pat_Age and that should give the sum of all 10 patients */
16 • select sum(Pat_Age) from Patients;
17 /* 3.3 AVG FUNCTION , We will calculate the average of Pat_Age and that should add all 10 ages and divide it by 10 */
18 • select avg(Pat_Age) from Patients;
```

The results grid shows the output of the query:

avg(Pat_Age)
50.1000

The output section shows the execution log:

#	Time	Action	Message
4	18:29:19	select count(Pat_Id) from Patients group by Pat_Sex LIMIT 0, 1000	Error Code: 1146. Table 'myhospital.pat_Id' doesn't exist
5	18:29:50	select count(Pat_Id) from Patients group by Pat_Sex LIMIT 0, 1000	2 row(s) returned
6	18:30:32	select count(Doc_Id) from Doctors group by Doc_Sex having count(Doc_Id)>5 LIMIT 0, 1000	2 row(s) returned
7	18:43:54	select count(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
8	18:46:59	select sum(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
9	18:49:28	select avg(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned

3.4 MIN Function

The screenshot shows a query editor with a SQL query and its results. The query is as follows:

```
15 /* 3.2 SUM FUNCTION , We will calculate the sum of Pat_Age and that should give the sum of all 10 patients */
16 • select sum(Pat_Age) from Patients;
17 /* 3.3 AVG FUNCTION , We will calculate the average of Pat_Age and that should add all 10 ages and divide it by 10 */
18 • select avg(Pat_Age) from Patients;
19 /* 3.4 MIN FUNCTION , We will calculate the minimum of Pat_Age and that should give the age of youngest patient */
20 • select min(Pat_Age) from Patients;
```

The results grid shows the output of the query:

min(Pat_Age)
25

The output section shows the execution log:

#	Time	Action	Message
5	18:29:50	select count(Pat_Id) from Patients group by Pat_Sex LIMIT 0, 1000	2 row(s) returned
6	18:30:32	select count(Doc_Id) from Doctors group by Doc_Sex having count(Doc_Id)>5 LIMIT 0, 1000	2 row(s) returned
7	18:43:54	select count(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
8	18:46:59	select sum(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
9	18:49:28	select avg(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
10	18:50:57	select min(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned

3.5 MAX Function

The screenshot shows a database query editor with the following SQL queries:

```
17 /* 3.3 AVG FUNCTION , We will calculate the average of Pat_Age and that should add all 10 ages and divide it by 10 */
18 select avg(Pat_Age) from Patients;
19 /* 3.4 MIN FUNCTION , We will calculate the minimum of Pat_Age and that should give the age of youngest patient */
20 select min(Pat_Age) from Patients;
21 /* 3.5 MAX FUNCTION , We will calculate the maximum of Pat_Age and that should give the age of oldest patient */
22 select max(Pat_Age) from Patients;
```

The results are displayed in a table:

max(Pat_Age)
85

The output section shows the following actions:

#	Time	Action	Message
6	18:38:32	select count(Doc_Id) from Doctors group by Doc_Sex having count(Doc_Id)>5 LIMIT 0, 1000	2 row(s) returned
7	18:43:54	select count(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
8	18:46:59	select sum(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
9	18:49:28	select avg(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
10	18:50:57	select min(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
11	18:52:26	select max(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned

4. Logical Operators especially with LIKE

4.1 Names of Patients starting with 'A'

The screenshot shows a database query editor with the following SQL queries:

```
20 select min(Pat_Age) from Patients;
21 /* 3.5 MAX FUNCTION , We will calculate the maximum of Pat_Age and that should give the age of oldest patient */
22 select max(Pat_Age) from Patients;
23 /* 4. LOGICAL OPERATORS WITH LIKE */
24 /* We will two examples of Pat_FName starting with 'A' and ending with 'A' and we will give all details of these people */
25 select * from Patients where Pat_FName like 'a%';
```

The results are displayed in a table:

Pat_Id	Pat_FName	Pat_LName	Pat_Disease	Pat_Age	Pat_Sex	Doc_Id
301	Abdul	Mohammed	Heart Disease	70	M	208
302	Abhinav	Singh	AutoImmune Disease	45	M	211
306	Anushka	Sharma	Stroke	85	F	208

The output section shows the following actions:

#	Time	Action	Message
7	18:43:54	select count(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
8	18:46:59	select sum(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
9	18:49:28	select avg(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
10	18:50:57	select min(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
11	18:52:26	select max(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
12	18:56:57	select * from Patients where Pat_FName like 'a%'; LIMIT 0, 1000	3 row(s) returned

4.2 Names of Patients ending with 'A'

The screenshot shows a SQL IDE with a query editor at the top and a results grid below. The query is as follows:

```
21 /* 3.5 MAX FUNCTION , We will calculate the maximum of Pat_Age and that should give the age of oldest patient */
22 select max(Pat_Age) from Patients;
23 /* 4. LOGICAL OPERATORS WITH LIKE */
24 /* We will two examples of Pat_FName starting with 'A' and ending with 'A' and we will give all details of these people */
25 select * from Patients where Pat_FName like 'a%';
26 select * from Patients where Pat_FName like '%a';
```

The results grid displays the following data:

Pat_Id	Pat_FName	Pat_LName	Pat_Disease	Pat_Age	Pat_Sex	Doc_Id
305	Anushka	Sharma	Stroke	85	F	208
308	Charita	N	Asthma	45	F	201
309	Garima	Singh	Diabetes	40	F	206
310	Nikita	Patil	Neural Disease	50	F	220

Below the results grid, the 'Output' tab shows the execution log:

#	Time	Action	Message
8	18:46:59	select sum(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
9	18:49:28	select avg(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
10	18:50:57	select min(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
11	18:52:26	select max(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned
12	18:56:57	select * from Patients where Pat_FName like 'a%' LIMIT 0, 1000	3 row(s) returned
13	18:57:45	select * from Patients where Pat_FName like '%a' LIMIT 0, 1000	4 row(s) returned

5. Nested Queries

5.1 First Nested Query with only one nested query

The screenshot shows a SQL IDE with a query editor at the top and a results grid below. The query is as follows:

```
24 /* We will two examples of Pat_FName starting with 'A' and ending with 'A' and we will give all details of these people */
25 select * from Patients where Pat_FName like 'a%';
26 select * from Patients where Pat_FName like '%a';
27 /* 5. NESTED QUERIES ATLEAST 4 */
28 /* 5.1 Using 1 nested query , it should give all details as all Pat_Id is greater than 100 */
29 select * from Patients where Pat_Id in (select Pat_Id from Bills where Pat_Id > 100);
```

The results grid displays the following data:

Pat_Id	Pat_FName	Pat_LName	Pat_Disease	Pat_Age	Pat_Sex	Doc_Id
301	Abdul	Muhammed	Heart Disease	70	M	208
302	Abhinav	Singh	Autoimmune Disease	45	M	211
303	Champak	Chacha	Infectious Disease	31	M	211
304	Kabir	Singh	Allergies	25	M	201
305	Rishan	Baba	Cancer	28	M	206
306	Anushka	Sharma	Stroke	85	F	208
307	Bhavani	Patil	Heart Disease	82	F	218
308	Charita	N	Asthma	45	F	201
309	Garima	Singh	Diabetes	40	F	206
310	Nikita	Patil	Neural Disease	50	F	220

Below the results grid, the 'Output' tab shows the execution log:

#	Time	Action	Message	Duration / Fetch
10	18:50:57	select max(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
11	18:52:26	select max(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
12	18:56:57	select * from Patients where Pat_FName like 'a%' LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
13	18:57:45	select * from Patients where Pat_FName like '%a' LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
14	19:03:51	select * from Patients where Pat_Id in (select Pat_Id from Bills where Pat_Id > 100)	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
15	19:04:44	select * from Patients where Pat_Id in (select Pat_Id from Bills where Pat_Id > 100) LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

5.2 Second Nested Query with two nested queries

The screenshot shows a SQL IDE window with a query editor containing the following SQL code:

```
26 * select * from Patients where Pat_Name like 'a';
27 /*
28 5.1 Using 1 nested query , it should give all details as all Pat_Id is greater than 100 */
29 * select * from Patients where Pat_Id in (select Pat_Id from Bills where Pat_Id > 100);
30 /*
31 5.2 Using 2 nested queries , it gives all details of Doctors */
32 * select * from Doctors where Doc_Id in (select Doc_Id from Patients where Pat_Id in (select Pat_Id from Relatives where Pat_Id > 100));
```

The results grid shows a table with columns: Doc_Id, Doc_Name, Doc_Age, Doc_Sex, and Speciality_Id. The data is as follows:

Doc_Id	Doc_Name	Doc_Age	Doc_Sex	Speciality_Id
201	Aarav	52	M	101
206	Jai	62	M	106
208	Shaurya	52	M	108
211	Angeli	46	F	101
219	Ranga	44	F	108
220	Rakha	41	F	110

The output pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
11	18:52:26	select max(Pat_Age) from Patients LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
12	18:56:57	select * from Patients where Pat_Name like 'a' LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
13	18:57:45	select * from Patients where Pat_Name like 'a' LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
14	19:03:51	select * from Patients where Pat_Id in (select Pat_Id from Bills where Pat_Id > 100)	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
15	19:04:44	select * from Patients where Pat_Id in (select Pat_Id from Bills where Pat_Id > 100) LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
16	19:10:45	select * from Doctors where Doc_Id in (select Doc_Id from Patients where Pat_Id in (select Pat_Id from Relativ...	6 row(s) returned	0.109 sec / 0.000 sec
17	19:10:45	select * from Doctors where Doc_Id in (select Doc_Id from Patients where Pat_Id in (select Pat_Id from Relativ...	6 row(s) returned	0.109 sec / 0.000 sec

5.3 Third Nested Query with two nested queries

The screenshot shows a SQL IDE window with a query editor containing the following SQL code:

```
28 /*
29 5.1 Using 1 nested query , it should give all details as all Pat_Id is greater than 100 */
30 * select * from Patients where Pat_Id in (select Pat_Id from Bills where Pat_Id > 100);
31 /*
32 5.2 Using 2 nested queries , it gives all details of Doctors */
33 * select * from Doctors where Doc_Id in (select Doc_Id from Patients where Pat_Id in (select Pat_Id from Relatives where Pat_Id > 100));
34 /*
35 5.3 Using 2 nested queries , it gives Consultation_Fee from Bills Table */
36 * select Consultation_Fee from Bills where Pat_Id in (select Pat_Id from Patients where Pat_Id in (select Pat_Id from Relatives where Pat_Id > 100));
```

The results grid shows a table with columns: Consultation_Fee. The data is as follows:

Consultation_Fee
10000
10500
8000
5000
15000
8000
11000
12500
4000
13000

The output pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
12	18:56:57	select * from Patients where Pat_Name like 'a' LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
13	18:57:45	select * from Patients where Pat_Name like 'a' LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
14	19:03:51	select * from Patients where Pat_Id in (select Pat_Id from Bills where Pat_Id > 100)	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
15	19:04:44	select * from Patients where Pat_Id in (select Pat_Id from Bills where Pat_Id > 100) LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
16	19:10:45	select * from Doctors where Doc_Id in (select Doc_Id from Patients where Pat_Id in (select Pat_Id from Relativ...	6 row(s) returned	0.109 sec / 0.000 sec
17	19:15:15	select Consultation_Fee from Bills where Pat_Id in (select Pat_Id from Patients where Pat_Id in (select Pat_Id fr...	10 row(s) returned	0.000 sec / 0.000 sec

5.4 Fourth Nested Query with two nested queries

The screenshot shows a SQL IDE window with a query editor containing the following SQL code:

```
30 /*
31 5.2 Using 2 nested queries , it gives all details of Doctors */
32 * select * from Doctors where Doc_Id in (select Doc_Id from Patients where Pat_Id in (select Pat_Id from Relatives where Pat_Id > 100));
33 /*
34 5.3 Using 2 nested queries , it gives Consultation_Fee from Bills Table */
35 * select Consultation_Fee from Bills where Pat_Id in (select Pat_Id from Patients where Pat_Id in (select Pat_Id from Relatives where Pat_Id > 100));
36 /*
37 5.4 Using 2 nested queries , it gives Speciality_Name from Specialities Table */
38 * select Speciality_Name from Specialities where Speciality_Id in (select Speciality_Id from Doctors where Doc_Id in (select Doc_Id from Patient...
```

The results grid shows a table with columns: Speciality_Name. The data is as follows:

Speciality_Name
Allergy & Immunology
Cardiology
Family Medicine
Neurology

The output pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
13	18:57:45	select * from Patients where Pat_Name like 'a' LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
14	19:03:51	select * from Patients where Pat_Id in (select Pat_Id from Bills where Pat_Id > 100)	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
15	19:04:44	select * from Patients where Pat_Id in (select Pat_Id from Bills where Pat_Id > 100) LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
16	19:10:45	select * from Doctors where Doc_Id in (select Doc_Id from Patients where Pat_Id in (select Pat_Id from Relativ...	6 row(s) returned	0.109 sec / 0.000 sec
17	19:15:15	select Consultation_Fee from Bills where Pat_Id in (select Pat_Id from Patients where Pat_Id in (select Pat_Id fr...	10 row(s) returned	0.000 sec / 0.000 sec
18	19:18:30	select Speciality_Name from Specialities where Speciality_Id in (select Speciality_Id from Doctors where Doc_I...	4 row(s) returned	0.016 sec / 0.000 sec

**NOTE: ALL THE EXPLANATION IS GIVEN IN THE SCRIPT FILE FOR
EACH QUESTIONS WITH QUESTION NUMBERS IN THE COMMENTS.**