SQL Practice Project -2 submitted by Sujit Sonar:

School Ranking Analysis.

DESCRIPTION
Consider an institution that wants to store the students' details and their marks records to track their progress. The database would contain the students' information, marks of the students with the rank that can be viewed, updated, and evaluated for the performance evaluation.

Objective:
The design of the database helps to easily retrieve thousands of student records.

Task to be performed:

- Write a query to create a **students** table with appropriate data types for student id, student first name, student last name, class, and age where the student last name, student first name, and student id should be a **NOT NULL constraint**, and the student id should be in a **primary key**.

```
3 •  create table students
4    (student_id int not null primary key,
5    first_name varchar(225) not null,
6    last_name varchar(225) not null,
7    class varchar(100),
8    age int) engine = InnoDB;
9
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	studen...	int	NO	PRI	NULL	
2	first_n...	varchar(225)	NO		NULL	
3	last_na...	varchar(225)	NO		NULL	
4	class	varchar(100)	YES		NULL	
5	age	int	YES		NULL	

- Write a query to create a **marksheet** table that includes score, year, ranking, class, and student id.

```
12 •  create table marksheet
13    (score int,
14    year int,
15    class varchar(100),
16    ranking int,
17    s_id int) engine = InnoDB;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	score	int	YES		NULL	
2	year	int	YES		NULL	
3	class	varchar(100)	YES		NULL	
4	ranking	int	YES		NULL	
5	s_id	int	YES		NULL	

- Write a query to **insert** values in **students** and **marksheet** tables**.**

Inserting values into students table:

```
24 •  insert into SQL_basics.students
25    (student_id, first_name, last_name,class,age)
26    values
27    (1,'krishna','gee','10',18),
28    (2,'stephen','christ','10',17),
29    (3,'kailash','kumar','10',18),
30    (4,'ashish','jain','10',16),
31    (5,'khusbu','jain','10',17),
32    (6,'madhan','lal','10',16),
33    (7,'saurab','kothari','10',15),
34    (8,'vinesh','roy','10',14),
35    (9,'rishika','r','10',15),
36    (10,'sara','ryan','10',16),
37    (11,'rosy','kumar','10',16);
```

```
39 •  select * from students;
```

Result Grid | Filter Rows: | Edit:

#	student_id	first_name	last_name	class	age
1	1	krishna	gee	10	18
2	2	stephen	christ	10	17
3	3	kailash	kumar	10	18
4	4	ashish	jain	10	16
5	5	khusbu	jain	10	17
6	6	madhan	lal	10	16
7	7	saurab	kothari	10	15
8	8	vinesh	roy	10	14
9	9	rishika	r	10	15
10	10	sara	ryan	10	16
11	11	rosy	kumar	10	16

Inserting values into marksheet table:

```
insert into SQL_basics.marksheet
(score, year, class,ranking,s_id)
values
(989,2014,'10',1,1),
(454,2014,'10',10,2),
(880,2014,'10',4,3),
(870,2014,'10',5,4),
(720,2014,'10',7,5),
(670,2014,'10',8,6),
(900,2014,'10',3,7),
(540,2014,'10',9,8),
(801,2014,'10',6,9),
(420,2014,'10',11,10),
(970,2014,'10',2,11),
(720,2014,'10',11,12);
```

```
58 •  select * from marksheet;
59
```

esult Grid | Filter Rows:

#	score	year	class	ranking	s_id
	989	2014	10	1	1
	454	2014	10	10	2
	880	2014	10	4	3
	870	2014	10	5	4
	720	2014	10	7	5
	670	2014	10	8	6
	900	2014	10	3	7
	540	2014	10	9	8
	801	2014	10	6	9
0	420	2014	10	11	10
1	970	2014	10	2	11
2	720	2014	10	11	12

- Write a query to display student id and student first name from the student table if the **age is greater than or equal to 16** and the **student's last name is Kumar.**

```
60 •  select student_id, first_name, last_name, age from SQL_basics.students
61    where age >= 16
62    and last_name = 'kumar';
```

esult Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell

#	student_id	first_name	last_name	age
	3	kailash	kumar	18
	11	rosy	kumar	16
	NULL	NULL	NULL	NULL

- Write a query to display all the details from the marksheet table **if the score is between 800 and 1000.**

```
66 •  select * from SQL_basics.marksheet
67    where score between 800 and 1000;
```

esult Grid | Filter Rows: | Export:

#	score	year	class	ranking	s_id
	989	2014	10	1	1
	880	2014	10	4	3
	870	2014	10	5	4
	900	2014	10	3	7
	801	2014	10	6	9
	970	2014	10	2	11

- Write a query to display the marksheet details from the marksheet table by **adding 5 to the score** and by naming the **column** as **new score**.

```
70 •  select * , score + 5 as new_score from SQL_basics.marksheet;
71
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	score	year	class	ranking	s_id	new_score
1	989	2014	10	1	1	994
2	454	2014	10	10	2	459
3	880	2014	10	4	3	885
4	870	2014	10	5	4	875
5	720	2014	10	7	5	725
6	670	2014	10	8	6	675
7	900	2014	10	3	7	905
8	540	2014	10	9	8	545
9	801	2014	10	6	9	806
10	420	2014	10	11	10	425
11	970	2014	10	2	11	975
12	720	2014	10	11	12	725

- Write a query to display the marksheet table in **descending order of the score**.

```
73 •  select * from SQL_basics.marksheet
74    order by score desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cel

#	score	year	class	ranking	s_id
1	989	2014	10	1	1
2	970	2014	10	2	11
3	900	2014	10	3	7
4	880	2014	10	4	3
5	870	2014	10	5	4
6	801	2014	10	6	9
7	720	2014	10	7	5
8	720	2014	10	11	12
9	670	2014	10	8	6
10	540	2014	10	9	8
11	454	2014	10	10	2
12	420	2014	10	11	10

- Write a query to display details of the students whose **first name starts with a.**

```
76
77 •  select * from SQL_basics.students
78    where first_name like 'a%';
```

Result Grid | Filter Rows: | Edit:

#	student_id	first_name	last_name	class	age
1	4	ashish	jain	10	16
*	NULL	NULL	NULL	NULL	NULL

################################### END###################################