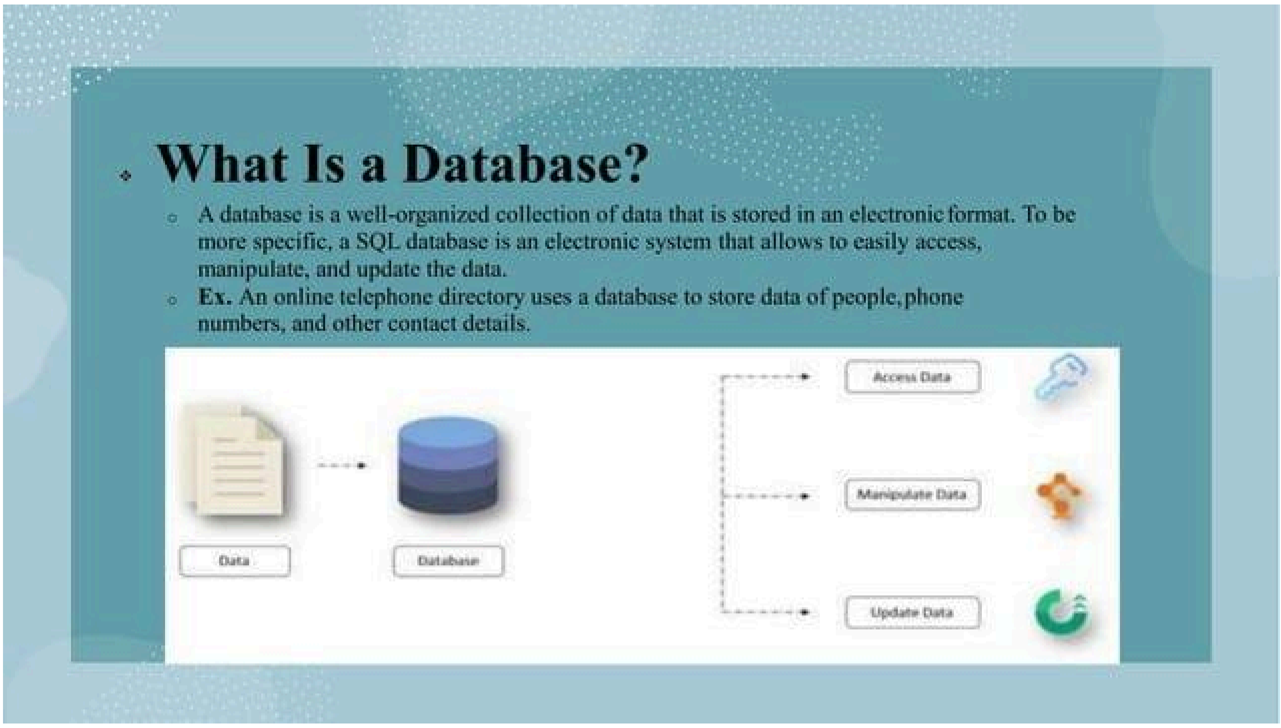
STRUCTURED QUERY LANGUAGE - SQL

CASE STUDY

RAJESH R 7/19/24 SQL

Introduction:



What Is a Database?

- A database is a well-organized collection of data that is stored in an electronic format. To be more specific, a SQL database is an electronic system that allows to easily access, manipulate, and update the data.
- **Ex.** An online telephone directory uses a database to store data of people, phone numbers, and other contact details.

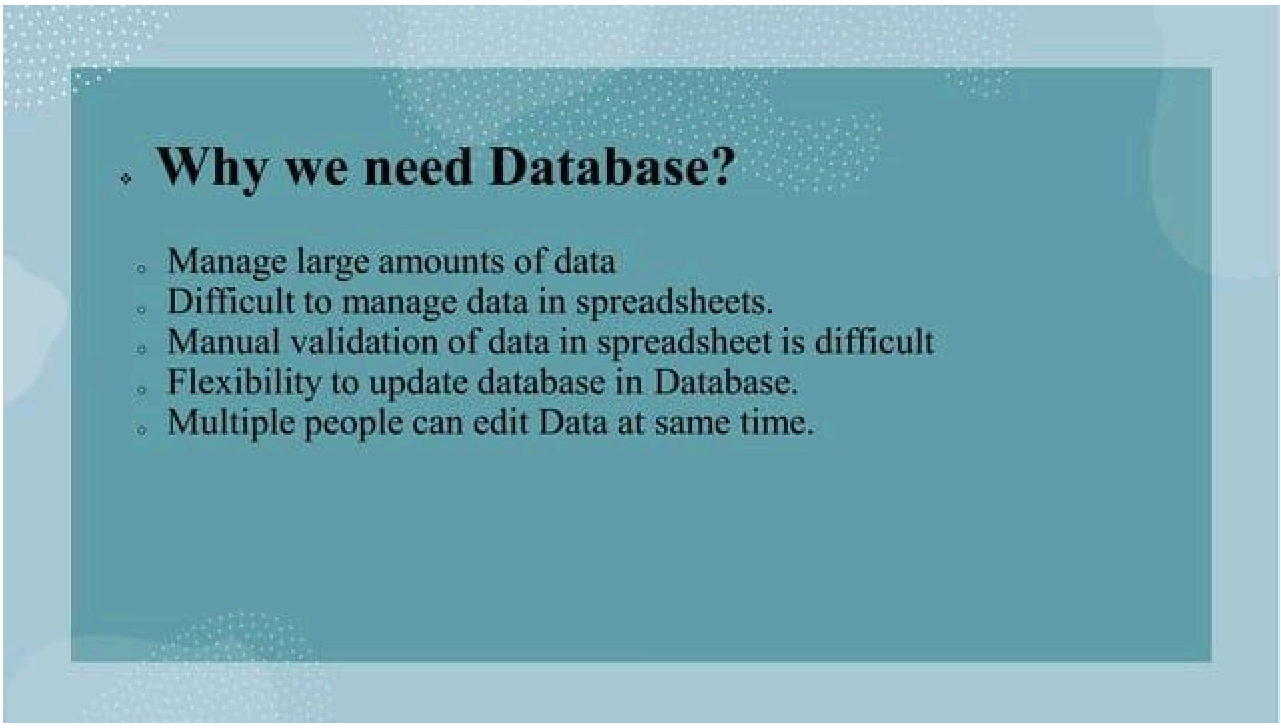
Data

Database

Access Data

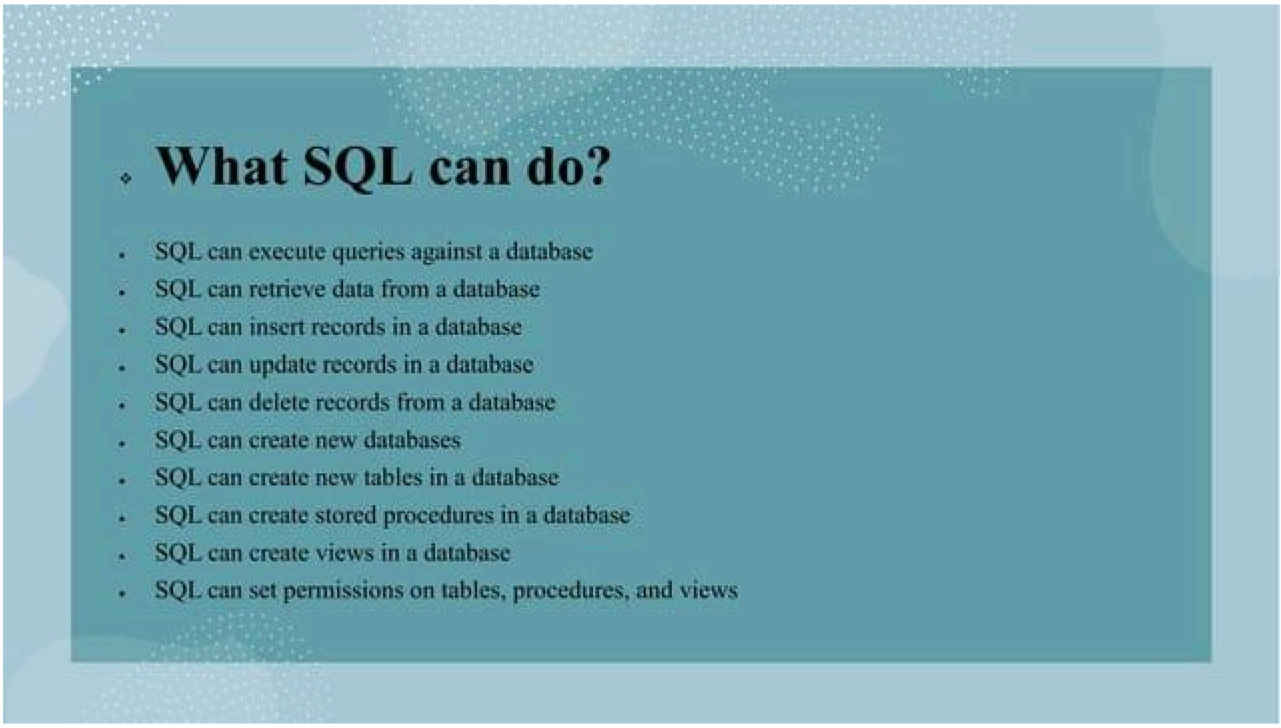
Manipulate Data

Update Data



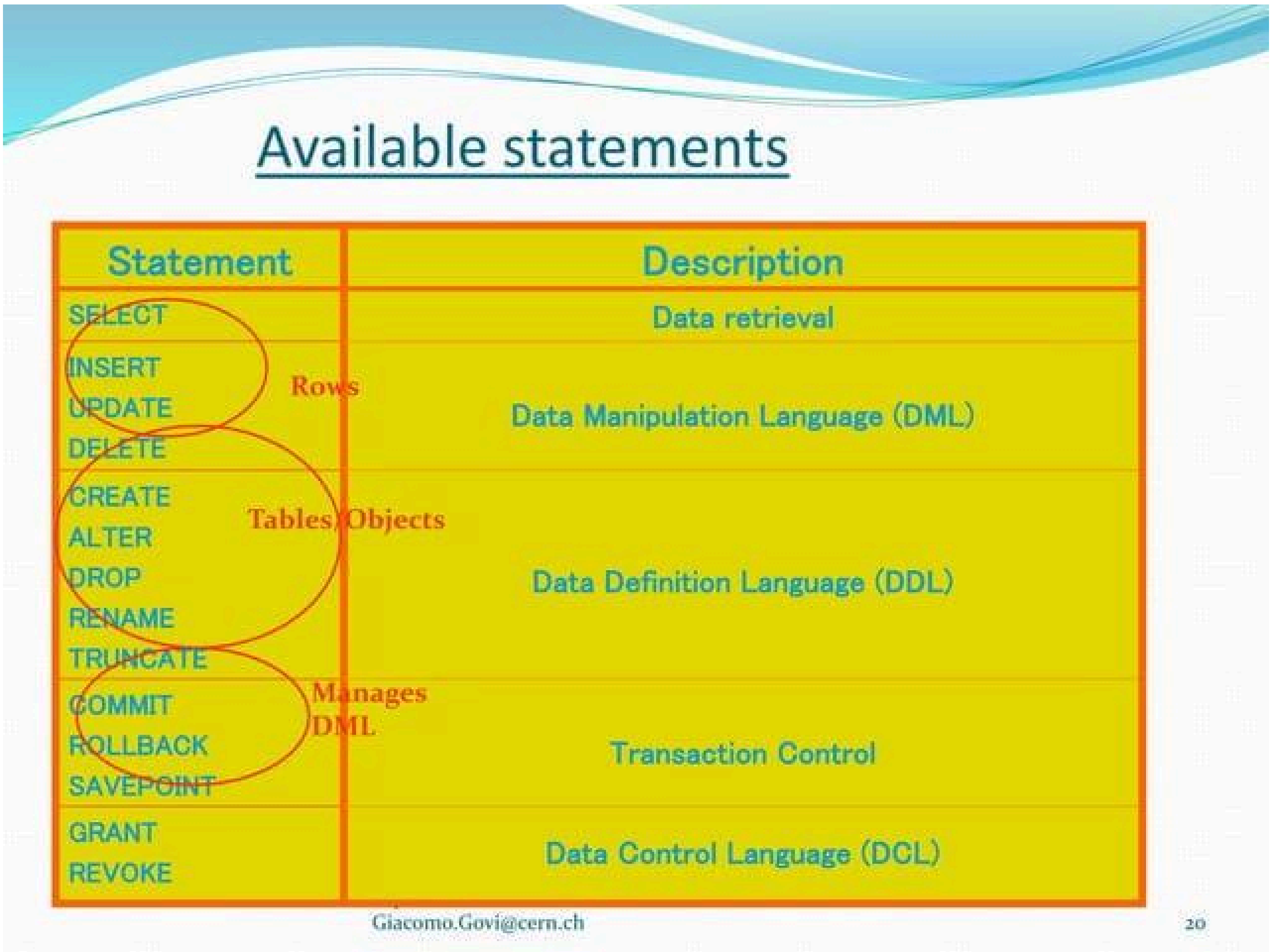
Why we need Database?

- Manage large amounts of data
- Difficult to manage data in spreadsheets.
- Manual validation of data in spreadsheet is difficult
- Flexibility to update database in Database.
- Multiple people can edit Data at same time.



❖ What SQL can do?

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views



Available statements

Statement	Description
SELECT	Data retrieval

INSERT
UPDATE
DELETE
	Data Manipulation Language (DML)

CREATE ALTER DROP RENAME TRUNCATE	Data Definition Language (DDL)
COMMIT ROLLBACK SAVEPOINT	Transaction Control
GRANT REVOKE	Data Control Language (DCL)

Rows

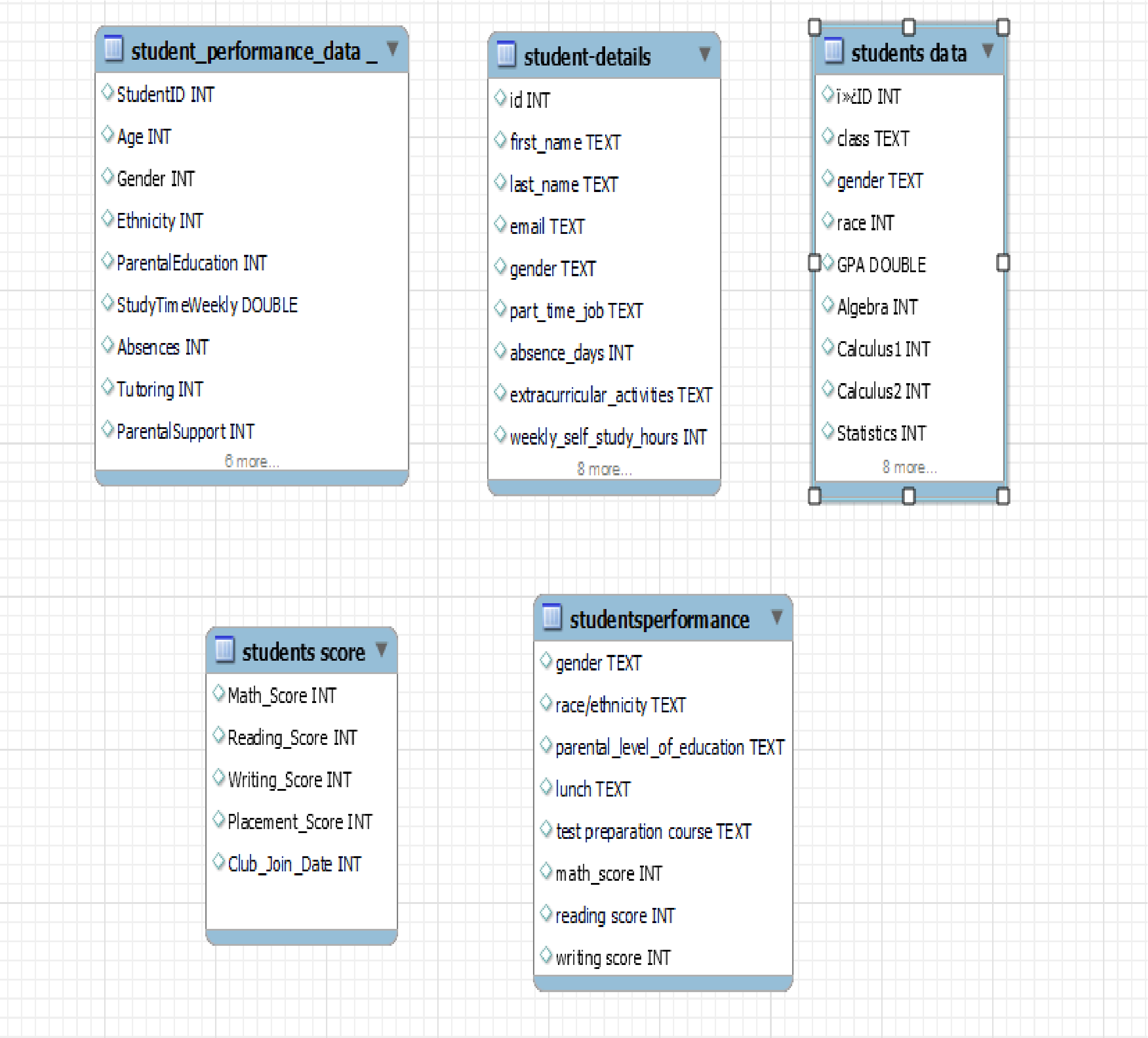
Tables/Objects

Manages DML

Giacomo.Govi@cern.ch

20

ER Diagram:



student_performance_data _

- StudentID INT
- Age INT
- Gender INT
- Ethnicity INT
- ParentalEducation INT
- StudyTimeWeekly DOUBLE
- Absences INT
- Tutoring INT
- ParentalSupport INT
- 6 more...

student-details

- id INT
- first_name TEXT
- last_name TEXT
- email TEXT
- gender TEXT
- part_time_job TEXT
- absence_days INT
- extracurricular_activities TEXT
- weekly_self_study_hours INT
- 8 more...

students data

- ï»¿ID INT
- class TEXT
- gender TEXT
- race INT
- GPA DOUBLE
- Algebra INT
- Calculus1 INT
- Calculus2 INT
- Statistics INT
- 8 more...

students score

- Math_Score INT
- Reading_Score INT
- Writing_Score INT
- Placement_Score INT
- Club_Join_Date INT

studentsperformance

- gender TEXT
- race/ethnicity TEXT
- parental_level_of_education TEXT
- lunch TEXT
- test preparation course TEXT
- math_score INT
- reading score INT
- writing score INT

Creating Tables:

Student data table

```
ID,class,gender,race,GPA,Algebra,Calculus1,Calculus2,Statistics,Probability,Measure,Functional_analysis,from1,from2,from3,from4,y
1141,A,male,1,73.47,64,81,87,60,74,71,60,A,A,A,3,0
1142,A,female,1,71.22,57,50,51,51,55,62,61,B,A,A,2,0
1143,A,female,2,74.56,47,48,71,60,61,68,64,C,A,A,0,1
1144,A,female,1,72.89,46,72,38,60,29,54,51,D,A,A,0,0
1145,A,female,1,70.11,49,45,63,60,66,66,61,E,A,A,0,0
1146,A,male,3,65.04,60,60,39,61,65,74,60,F,B,C,0,0
1147,A,male,4,77.11,60,43,52,63,71,72,75,G,A,A,0,1
1148,A,female,5,64.75,60,38,60,63,70,68,51,H,B,C,0,0
1149,B,female,5,77.92,61,60,66,68,80,78,71,I,B,A,0,0
1150,A,female,5,76.5,60,61,60,69,73,60,62,H,B,A,0,0
1151,A,male,1,76.83,61,66,61,69,65,69,67,H,B,A,0,0
1152,B,female,1,83.65,73,75,76,71,85,74,89,J,B,A,2,0
1153,B,female,1,77,70,60,71,71,87,68,61,K,B,D,0,0
1154,A,male,1,80.08,67,65,74,73,87,68,76,E,B,A,0,2
1155,A,male,1,81.67,73,60,55,76,77,74,64,L,A,A,0,2
1156,B,female,6,82.07,72,63,70,76,81,86,73,E,B,A,0,2
1157,A,male,1,74.11,74,70,61,76,68,57,61,M,B,A,0,2
1158,A,male,1,81.5,68,67,55,76,85,75,87,B,A,A,0,0
1159,B,male,1,75.64,60,68,54,76,76,76,64,N,B,A,0,0
1160,A,male,7,76.46,67,73,64,78,83,76,64,O,B,B,0,2
1161,B,male,1,75.57,63,64,71,78,69,76,62,I,B,A,0,2
1162,B,male,1,83.13,82,82,92,78,92,83,78,N,B,A,2,0
1163,B,female,1,79.7,72,68,55,78,77,68,71,B,B,D,2,0
1164,A,female,1,83.98,71,73,83,79,88,87,81,F,B,A,2,0
1165,A,male,3,79.56,64,62,76,80,81,80,74,F,B,C,0,0
1166,A,female,4,78.54,76,67,71,80,71,83,67,P,A,D,0,0
1167,B,female,1,77.76,67,46,71,80,82,69,64,P,B,A,2,0
1168,A,female,1,83.06,68,72,71,81,86,81,70,F,B,A,2,0
1169,A,male,1,84.04,73,80,85,83,79,73,62,Q,B,S,0,0
1170,A,female,1,85.29,77,64,64,83,95,80,88,R,A,A,0,2
1171,B,female,1,85.04,76,75,85,83,88,82,69,J,B,A,2,0
1172,A,male,3,87.04,87,98,93,83,95,87,79,B,B,B,0,0
1173,A,male,2,74.17,63,73,61,83,72,55,62,C,A,A,0,0
```

Student details table

```
id,first_name,last_name,email,gender,part_time_job,absence_days,extracurricular_activities,weekly_self_study_hours,career_aspiration,math_score,history_score,physics_score,chemistry_score,biology_score,english_score,geography_score
1,Paul,Casey,paul.casey.1@gslingacademy.com,male,False,3,False,27,Lawyer,73,81,93,97,63,80,87
2,Danielle,Sandoval,danielle.sandoval.2@gslingacademy.com,female,False,2,False,47,Doctor,90,86,96,100,90,88,90
3,Tina,Andrews,tina.andrews.3@gslingacademy.com,female,False,9,True,13,Government Officer,81,97,95,96,65,77,94
4,Tara,Clark,tara.clark.4@gslingacademy.com,female,False,5,False,3,Artist,71,74,88,80,89,63,86
5,Anthony,Campos,anthony.campos.5@gslingacademy.com,male,False,5,False,10,Unknown,84,77,65,65,80,74,76
6,Kelly,Wade,kelly.wade.6@gslingacademy.com,female,False,2,False,26,Unknown,93,100,67,78,72,80,84
7,Anthony,Smith,anthony.smith.7@gslingacademy.com,male,False,3,True,23,Software Engineer,99,96,97,73,88,76,64
8,George,Short,george.short.8@gslingacademy.com,male,True,2,True,34,Software Engineer,95,95,82,63,84,70,85
9,Stanley,Gutierrez,stanley.gutierrez.9@gslingacademy.com,male,False,6,False,25,Unknown,94,68,94,85,81,74,72
10,Audrey,Simpson,audrey.simpson.10@gslingacademy.com,female,False,3,True,18,Teacher,98,69,88,71,67,71,73
11,Gabrielle,White,gabrielle.white.11@gslingacademy.com,female,False,2,False,7,Teacher,65,60,97,94,71,81,66
12,Clinton,Randolph,clinton.randolph.12@gslingacademy.com,male,False,1,False,7,Unknown,80,61,100,65,87,64,61
13,Patricia,Gomez,patricia.gomez.13@gslingacademy.com,female,True,7,False,4,Business Owner,94,59,69,67,89,65,73
14,Pamela,Jackson,pamela.jackson.14@gslingacademy.com,female,False,10,False,2,Business Owner,66,94,86,100,57,90,63
15,Laura,Jackson,laura.jackson.15@gslingacademy.com,female,False,3,False,39,Doctor,96,90,86,92,92,95,87
16,Roger,Wiley,roger.wiley.16@gslingacademy.com,male,False,6,False,0,Business Owner,94,50,78,64,79,74,84
17,Vicki,Thompson,vicki.thompson.17@gslingacademy.com,female,False,3,True,30,Scientist,92,64,93,91,80,89,72
18,Maxwell,Davidson,maxwell.davidson.18@gslingacademy.com,male,False,2,True,28,Software Engineer,86,83,85,79,93,76,77
19,Jonathan,Werner,jonathan.werner.19@gslingacademy.com,male,False,1,False,37,Doctor,92,87,92,99,97,87,86
20,Angela,Rios,angela.rios.20@gslingacademy.com,female,False,2,False,27,Software Engineer,99,65,98,75,66,72,100
```

Student performance data table

```
StudentID,Age,Gender,Ethnicity,ParentalEducation,StudyTimeWeekly,Absences,Tutoring,ParentalSupport,Extracurricular,Sports,Music,Volunteering,GPA,GradeClass
1001,17,1,0,2,19.833722807854713,7,1,2,0,0,1,0,2.929195591667681,2.0
1002,18,0,0,1,15.40875605584674,0,0,1,0,0,0,0,3.042914833436377,1.0
1003,15,0,2,3,4.21056976881226,26,0,2,0,0,0,0,0.1126022544661815,4.0
1004,17,1,0,3,10.028829473958215,14,0,3,1,0,0,0,2.0542181397029484,3.0
1005,17,1,0,2,4.6724952729713305,17,1,3,0,0,0,0,1.2880611817953875,4.0
1006,18,0,0,1,8.191218545250186,0,0,1,1,0,0,0,3.0841836144863937,1.0
1007,15,0,1,1,15.601680474699295,10,0,3,0,1,0,0,2.748237414891583,2.0
1008,15,1,1,4,15.424496305808074,22,1,1,1,0,0,0,1.360142712316461,4.0
1009,17,0,0,0,4.562007558047703,1,0,2,0,1,0,1,2.896819189513569,2.0
1010,16,1,0,1,18.444466363097202,0,0,3,1,0,0,0,3.5734742103297656,0.0
1011,17,0,0,1,11.851363655296536,11,0,1,0,0,0,0,2.1471716250185144,3.0
1012,17,0,0,1,7.59848581924029,15,0,2,0,0,0,1,1.5595945190402773,4.0
1013,17,0,1,1,10.038711615617213,21,0,3,1,0,0,0,1.520077814874808,4.0
1014,17,0,1,2,12.101425068754875,21,0,4,0,1,0,0,1.7515809583340785,4.0
1015,18,1,0,1,11.197810636915708,9,1,2,0,0,0,0,2.396788117124796,3.0
1016,15,0,0,2,9.728100710723563,17,1,0,0,1,0,0,1.3415207165346672,4.0
1017,18,0,3,1,10.098656081788002,14,0,2,1,1,0,0,2.2321752777159762,3.0
1018,18,1,0,0,3.5282382085577235,16,1,2,0,0,0,0,1.3844041756940335,4.0
1019,18,0,1,3,16.25465808609359,29,0,2,1,0,0,1,0.4695533233798704,4.0
1020,17,0,0,1,10.835206398820308,9,0,2,0,0,1,0,2.3957840945306996,3.0
1021,16,1,0,3,2.621597234094062,2,0,3,0,0,0,1,2.778411299920653,2.0
1022,15,0,0,2,15.323142031655559,25,0,1,1,0,0,0,0.34689403670501484,4.0
1023,16,1,1,0,18.648879567547016,29,1,1,0,0,0,0,0.3125462305253549,4.0
1024,18,1,3,4,18.94613798473924,20,0,2,1,0,0,0,1.7701318767799732,4.0
1025,18,1,0,1,7.380354648223455,15,0,2,0,0,0,0,1.5051556220362858,4.0
1026,16,1,0,3,2.7103374712150807,5,0,4,0,0,1,0,2.977851918315743,2.0
1027,16,0,0,1,10.367992532661303,2,0,2,0,1,0,0,2.948717671911921,2.
```

Student score table

```
id,first_name,last_name,email,gender,part_time_job,absence_days,extracurricular_activities,weekly_self_study_hours,career_aspiration,math_score,history_score,physics_score,chemistry_score,biology_score,english_score,geography_score
1,Paul,Casey,paul.casey.1@gslingacademy.com,male,False,3,False,27,Lawyer,73,81,93,97,63,80,87
2,Danielle,Sandoval,danielle.sandoval.2@gslingacademy.com,female,False,2,False,47,Doctor,90,86,96,100,90,88,90
3,Tina,Andrews,tina.andrews.3@gslingacademy.com,female,False,9,True,13,Government Officer,81,97,95,96,65,77,94
4,Tara,Clark,tara.clark.4@gslingacademy.com,female,False,5,False,3,Artist,71,74,88,80,89,63,86
5,Anthony,Campos,anthony.campos.5@gslingacademy.com,male,False,5,False,10,Unknown,84,77,65,65,80,74,76
6,Kelly,Wade,kelly.wade.6@gslingacademy.com,female,False,2,False,26,Unknown,93,100,67,78,72,80,84
7,Anthony,Smith,anthony.smith.7@gslingacademy.com,male,False,3,True,23,Software Engineer,99,96,97,73,88,76,64
8,George,Short,george.short.8@gslingacademy.com,male,True,2,True,34,Software Engineer,95,95,82,63,84,70,85
9,Stanley,Gutierrez,stanley.gutierrez.9@gslingacademy.com,male,False,6,False,25,Unknown,94,68,94,85,81,74,72
10,Audrey,Simpson,audrey.simpson.10@gslingacademy.com,female,False,3,True,18,Teacher,98,69,88,71,67,71,73
11,Gabrielle,White,gabrielle.white.11@gslingacademy.com,female,False,2,False,7,Teacher,65,60,97,94,71,81,66
12,Clinton,Randolph,clinton.randolph.12@gslingacademy.com,male,False,1,False,7,Unknown,80,61,100,65,87,64,61
13,Patricia,Gomez,patricia.gomez.13@gslingacademy.com,female,True,7,False,4,Business Owner,94,59,69,67,89,65,73
14,Pamela,Jackson,pamela.jackson.14@gslingacademy.com,female,False,10,False,2,Business Owner,66,94,86,100,57,90,63
15,Laura,Jackson,laura.jackson.15@gslingacademy.com,female,False,3,False,39,Doctor,96,90,86,92,92,95,87
16,Roger,Wiley,roger.wiley.16@gslingacademy.com,male,False,6,False,0,Business Owner,94,50,78,64,79,74,84
17,Vicki,Thompson,vicki.thompson.17@gslingacademy.com,female,False,3,True,30,Scientist,92,64,93,91,80,89,72
18,Maxwell,Davidson,maxwell.davidson.18@gslingacademy.com,male,False,2,True,28,Software Engineer,86,83,85,79,93,76,77
19,Jonathan,Werner,jonathan.werner.19@gslingacademy.com,male,False,1,False,37,Doctor,92,87,92,99,97,87,86
20,Angela,Rios,angela.rios.20@gslingacademy.com,female,False,2,False,27,Software Engineer,99,65,98,75,66,72,100
```

Students performance table

```
"gender","race/ethnicity","parental level of education","lunch","test preparation course","math score","reading score","writing score"
"female","group B","bachelor's degree","standard","none","72","72","74"
"female","group C","some college","standard","completed","69","90","88"
"female","group B","master's degree","standard","none","90","95","93"
"male","group A","associate's degree","free/reduced","none","47","57","44"
"male","group C","some college","standard","none","76","78","75"
"female","group B","associate's degree","standard","none","71","83","78"
"female","group B","some college","standard","completed","88","95","92"
"male","group B","some college","free/reduced","none","40","43","39"
"male","group D","high school","free/reduced","completed","64","64","67"
"female","group B","high school","free/reduced","none","38","60","50"
"male","group C","associate's degree","standard","none","58","54","52"
"male","group D","associate's degree","standard","none","40","52","43"
"female","group B","high school","standard","none","65","81","73"
"male","group A","some college","standard","completed","78","72","70"
"female","group A","master's degree","standard","none","50","53","58"
"female","group C","some high school","standard","none","69","75","78"
"male","group C","high school","standard","none","88","89","86"
"female","group B","some high school","free/reduced","none","18","32","28"
"male","group C","master's degree","free/reduced","completed","46","42","46"
"female","group C","associate's degree","free/reduced","none","54","58","61"
"male","group D","high school","standard","none","66","69","63"
"female","group B","some college","free/reduced","completed","65","75","70"
```

Datasets:

```
Select * from students performance;

 Select * from `students data`;

 Select * from `students performance data`;

 Select * from students details;

 Select * from students score;
```

Functions :

COUNT(): Returns the count of rows in a table or group.
SUM(): Calculates the sum of values in a column.
AVG(): Computes the average value for a specified column.
MAX(): Retrieves the maximum value in a column.
MIN(): Finds the minimum value in a column.

#To find math score maximum and minimum values from students score

```
Select  max(math_score),min(math_score) from `students score`;
```

#To find sum of all writing score fields in the students score table

```
Select sum (writing_score)from`students score`;
```

#To find avg of all placement score fields in the students score table

```
Select avg(placement_score)from`students score`;
```

#To find total number of club joining date in the students score table

```
Select count (club_join_date)from `students score`;
```

Operators:

An operator is a reserved word or a character that is used to query our database in a SQL expression.

- Arithmetic Operators
- Bitwise operators
- Logical operators
- Comparison operator
- Compound operators

Arithmetic Operators:

+	[Addition]
-	[Subtraction]
/	[Division]
*	[Multiplication]
%	[Modulus]

Bitwise operators:

&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR

Logical operators:

ALL - TRUE if all of the subquery values meet the condition

AND - TRUE if all the conditions separated by AND is TRUE

BETWEEN - TRUE if the operand is within the range of comparisons

IN - TRUE if the operand is equal to one of a list of expressions

LIKE - TRUE if the operand matches a pattern

NOT - Displays a record if the condition(s) is NOT TRUE

OR - TRUE if any of the conditions separated by OR is TRUE

- <u>Comparison operator :</u>

=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
<>	Not equal to

- <u>Compound operator:</u>

+=	Add equals
-=	Subtract equals
*=	Multiply equals
/=	Divide equals
%=	Modulo equals
&=	Bitwise AND equals
^-=	Bitwise exclusive equal

Example queries and answers:

1) To find list of students performance data with the following students Id and age : (1001, 17)

Select StudentID,age from `student_performance_data _` where StudentID=1001 and age=17;

2) To Find the list of students performance data same age(17) in different students Id:

Select StudentID,age from `student_performance_data _` where StudentID=1001 or age=17;

3)To Find the all the values student performance data in particular columns: (1001,1002)

Select * from `student_performance_data _` where StudentID in (1001,1002);

4) To find the values of expect those columns with the students Id from table:(1003,1004)

```
Select * from `student_performance_data _` where StudentID not in (1003,1004);
```

5) To find the first name starting with ‘P’ from the student details table:

```
Select first_name from`student-details` where first_name like 'P%';
```

6) To find the join date for students in range of ‘2019 to 2021’ from students score table:

```
Select * from `students score` where Club_Join_Date between 2019 and 2021;
```

7) To find the full name from students in students details table:

```
Select concat(first_name,' ',last_name)as full_name from `student-details`;
```

8) To find the total math and reading score of the students from students score table:

```
Select round(math_score,reading_score)from`students score`;
```

9) Return the list alphabetical wise for first name and last name from students details table:

```
Select first_name,last_name from `student-details` order by first_name,last_name;
```

10) Return the unique students names from students details table:

```
Select first_name,last_name from `student-details` group by first_name,last_name;
```

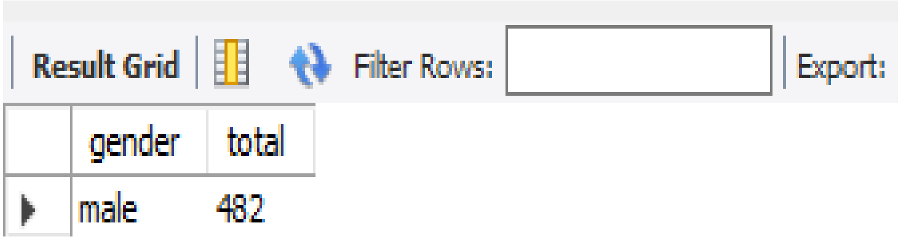
Case study Question & answers

11) write an sql query to fetch the count of table having gender “Male” in students performance table

*select * from studentsperformance;*

```
SELECT
        gender, COUNT(*) AS total
    FROM
        studentsperformance
     WHERE
        gender = 'male';
```

Output :



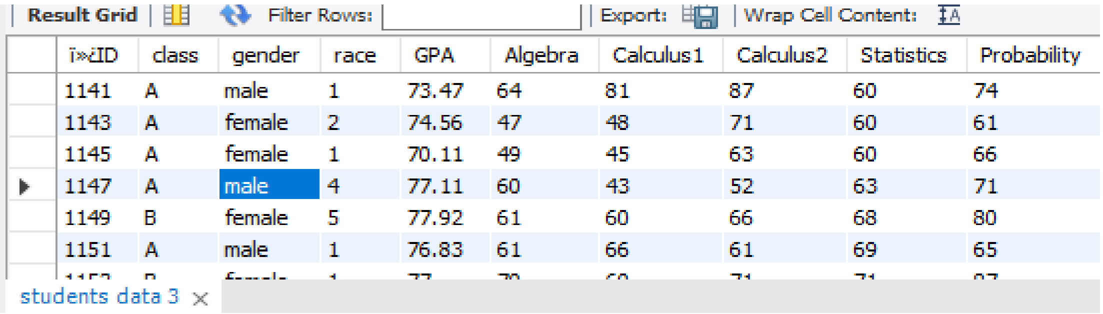
gender	total
male	482

12) write an sql query to show only odd rows in the student data table?

Select * from `students data`;

```
    select *
      FROM
         `students data`
  WHERE
         student_ID % 2 ;
```

Output:

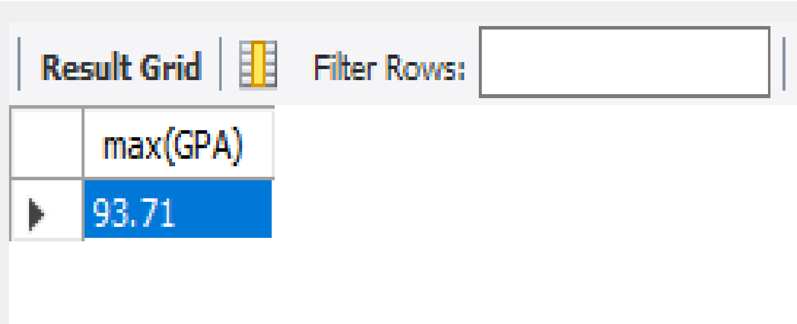


ï»¿ID	class	gender	race	GPA	Algebra	Calculus1	Calculus2	Statistics	Probability
1141	A	male	1	73.47	64	81	87	60	74
1143	A	female	2	74.56	47	48	71	60	61
1145	A	female	1	70.11	49	45	63	60	66
1147	A	male	4	77.11	60	43	52	63	71
1149	B	female	5	77.92	61	60	66	68	80
1151	A	male	1	76.83	61	66	61	69	65

13) write an SQL query to show the second highest GPA from a student data table using subquery?

```
select * from `students data`;

SELECT
        MAX(GPA)
    FROM
        `students data`
    WHERE
        GPA NOT IN (SELECT  MAX(GPA)
     FROM
        `students data`);
```

Output:



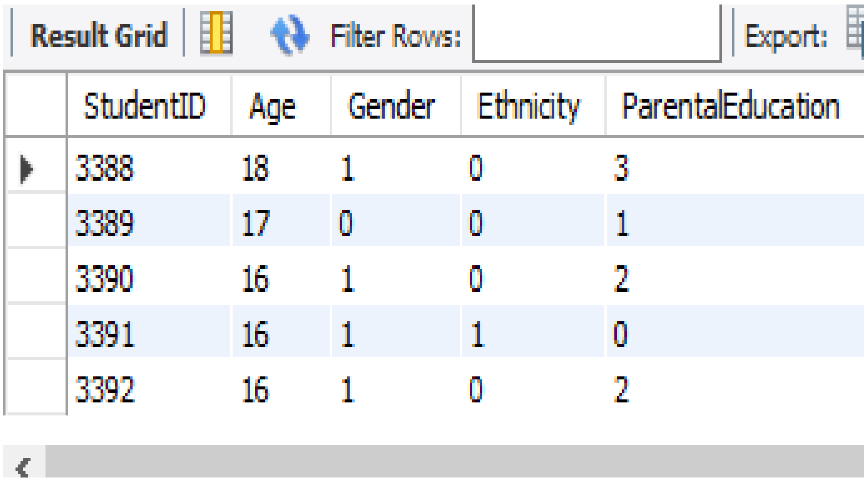
max(GPA)
93.71

14) write an sql query to fetch the last five records in orderwise from the students performance data table?

*select*from `student_performance_data _`;*

```
SELECT
    *
FROM
    (SELECT
        *
    FROM
        `student_performance_data _`
ORDER BY (StudentID) DESC
LIMIT 5) AS abc
ORDER BY StudentID;
```

Output:

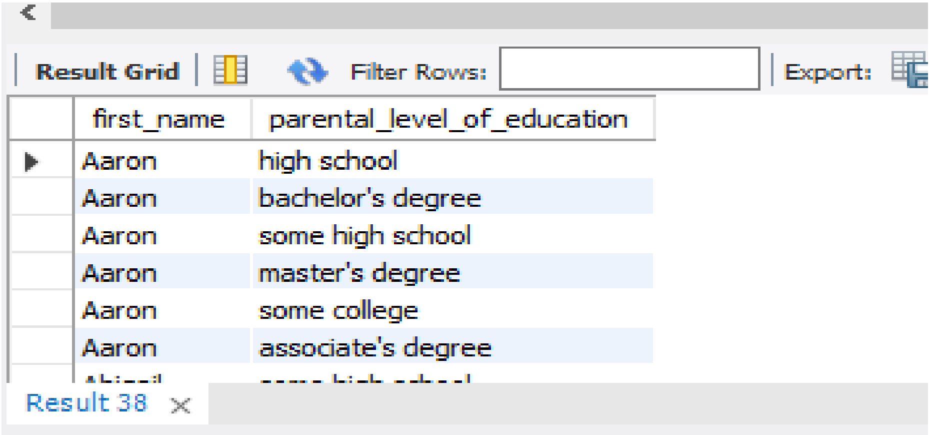


StudentID	Age	Gender	Ethnicity	ParentalEducation
3388	18	1	0	3
3389	17	0	0	1
3390	16	1	0	2
3391	16	1	1	0
3392	16	1	0	2

15) Get the unique first name and level of education order by first name from students detail and performance table

```
select * from  studentsperformance;

select * from  `student-details`;

  SELECT

 distinct first_name,parental_level_of_education

  FROM

      studentsperformance s

  left outer join

 `student-details` p ON s.gender = p.gender

  order by first_name;
```

Output:



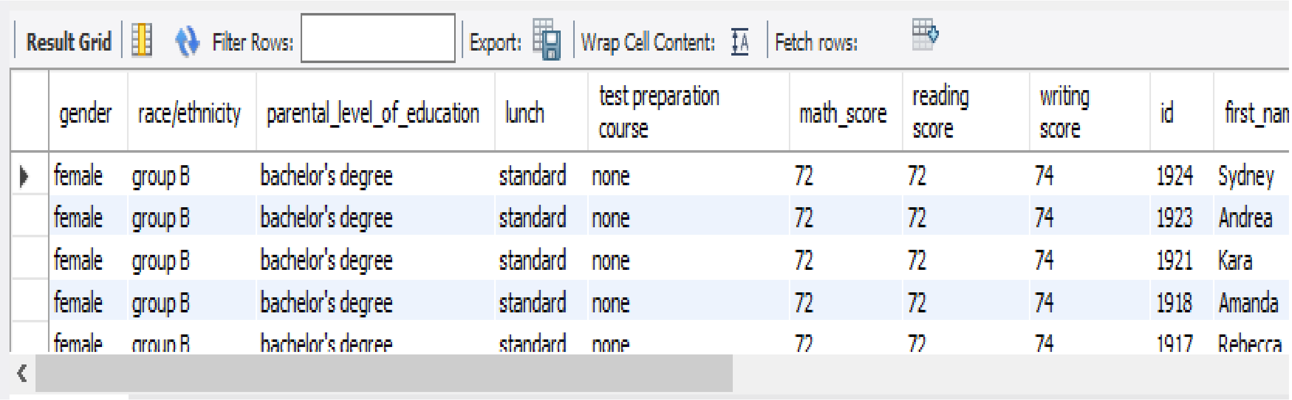
first_name	parental_level_of_education
Aaron	high school
Aaron	bachelor's degree
Aaron	some high school
Aaron	master's degree
Aaron	some college
Aaron	associate's degree
Abigail	some high school

16) How to get all the values of combining two tables in join funtions?

```
select * from  studentsperformance;

select * from  `student-details`;
```

```
            SELECT
                *
             FROM
        studentsperformance s
        Left join
       `student-details` p ON s.gender = p.gender
UNION ALL
        SELECT
               *
           FROM
         studentsperformance s
        right join
       `student-details` p ON s.gender = p.gender;
```

Output:

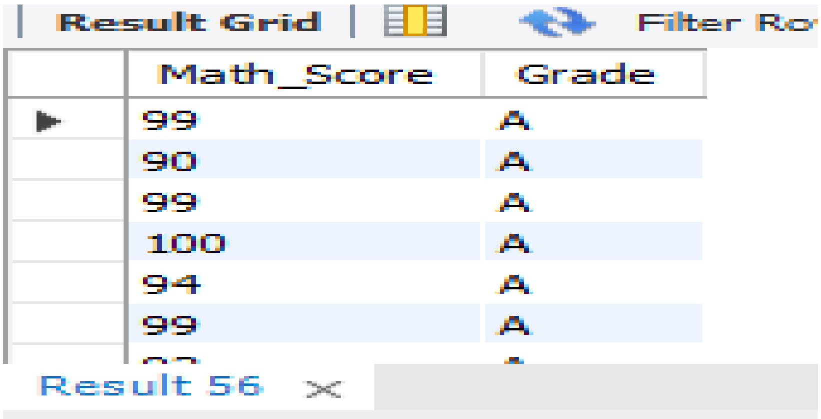


gender	race/ethnicity	parental_level_of_education	lunch	test preparation course	math_score	reading score	writing score	id	first_nan
female	group B	bachelor's degree	standard	none	72	72	74	1924	Sydney
female	group B	bachelor's degree	standard	none	72	72	74	1923	Andrea
female	group B	bachelor's degree	standard	none	72	72	74	1921	Kara
female	group B	bachelor's degree	standard	none	72	72	74	1918	Amanda
female	group B	bachelor's degree	standard	none	72	72	74	1917	Rebecca

17) Consider a students score table with columns math score.we want to categorise students into grades : A(90 to 100) , B(80 to 89), C(70 to 79), D(60 to 69), F(<60) use case expressions.

```
    select * from `students score`;

  SELECT

        Math_Score,

  CASE

  WHEN Math_score >= 90 THEN 'A'

  WHEN Math_score >= 80 AND Math_score < 90 THEN 'B'

  WHEN Math_score >= 70 AND Math_score < 80 THEN 'C'

  WHEN Math_score >= 60 AND Math_score < 70 THEN 'D'

  ELSE 'F'

 END AS Grade

FROM

 `students score`

ORDER BY Grade;
```

Output:

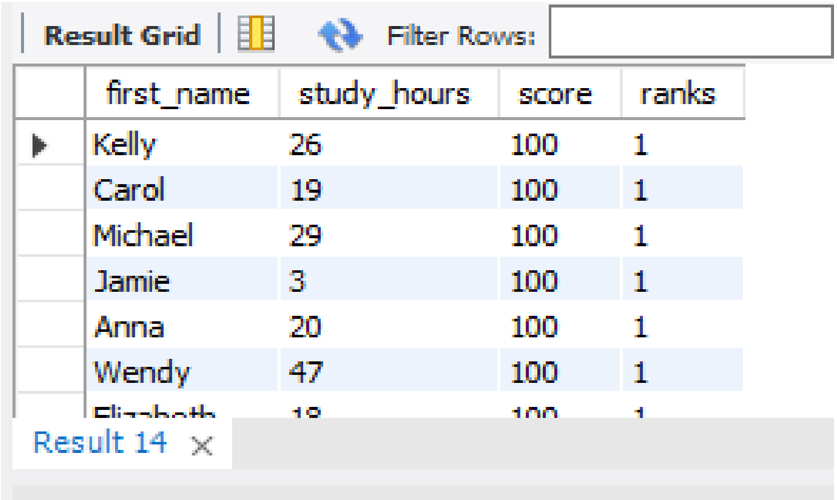


	Math_Score	Grade
▶	99	A
	90	A
	99	A
	100	A
	94	A
	99	A

18) Find rank the score in 100 students by their score in descending order from the students detail table?

```
                    select * from  `student-details`;

                         SELECT

                    first_name,weekly_self_study_hours as

       study_hours,history_score as score,

                         dense_rank()

                          over(order by history_score desc)as ranks

                          FROM`student-details`

                          limit 100;
```

Output:



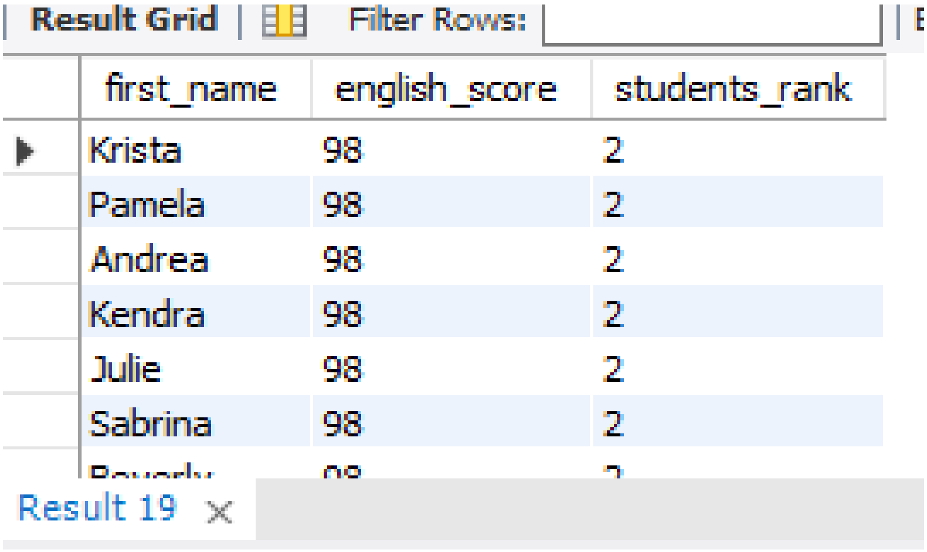
first_name	study_hours	score	ranks
Kelly	26	100	1
Carol	19	100	1
Michael	29	100	1
Jamie	3	100	1
Anna	20	100	1
Wendy	47	100	1
Elizabeth	18	100	1

19) Write a sql query to get the details of students whose marks are second highest in a particular english subject using the students details table?

```
select * from  `student-details`;
```

```
with CTE as(
SELECT
   first_name,english_score,
dense_rank()
over(partition by gender order by english_score desc) as
students_rank
FROM `student-details`)
SELECT
*
FROM CTE
WHERE students_rank = 2;
```

Output:



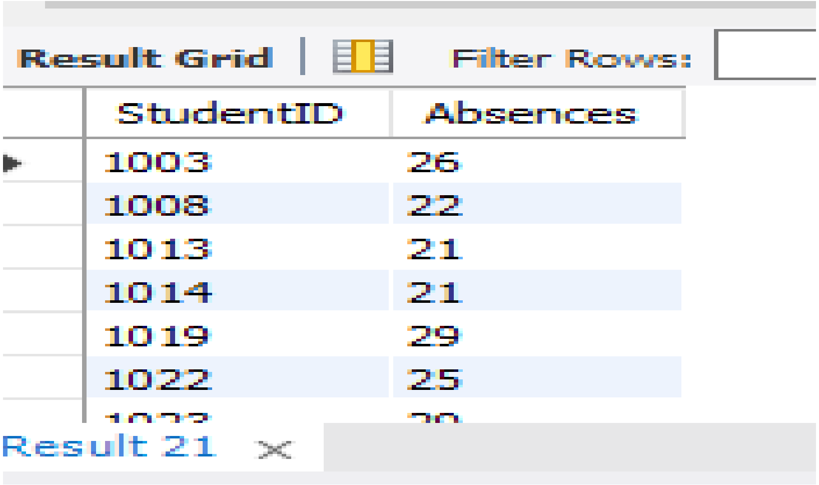
first_name	english_score	students_rank
Krista	98	2
Pamela	98	2
Andrea	98	2
Kendra	98	2
Julie	98	2
Sabrina	98	2
Beverly	98	2

20) Find the student's number of absences greater than 20 days to fetch the student id and absences from students performance data table?

*select*from `student_performance_data _;*

```
WITH `student_performance_data _`
    (StudentID,Absences) as (
  SELECT StudentID,sum(absences)
  FROM `student_performance_data _`
   group by StudentID
    )
   SELECT * FROM `student_performance_data _`
   WHERE absences > 20;
```

Output:



StudentID	Absences
1003	26
1008	22
1013	21
1014	21
1019	29
1022	25
1023	[illegible]

Conclusion:

The breadth and scope of the SQL commands provide the capability to create and manipulate a wide variety of database objects using the various CREATE, ALTER, and DROP commands. Those database objects then can be loaded with data using commands such as INSERT. The data can be manipulated using a wide variety of commands, such as SELECT, DELETE, and TRUNCATE, as well as the cursor commands, DECLARE, OPEN, FETCH, and CLOSE. Transactions to manipulate the data are controlled through the SET command, plus the COMMIT and ROLLBACK commands. And finally, other commands covered in this chapter include those that control a user's access to database resources through commands such as GRANT and REVOKE.