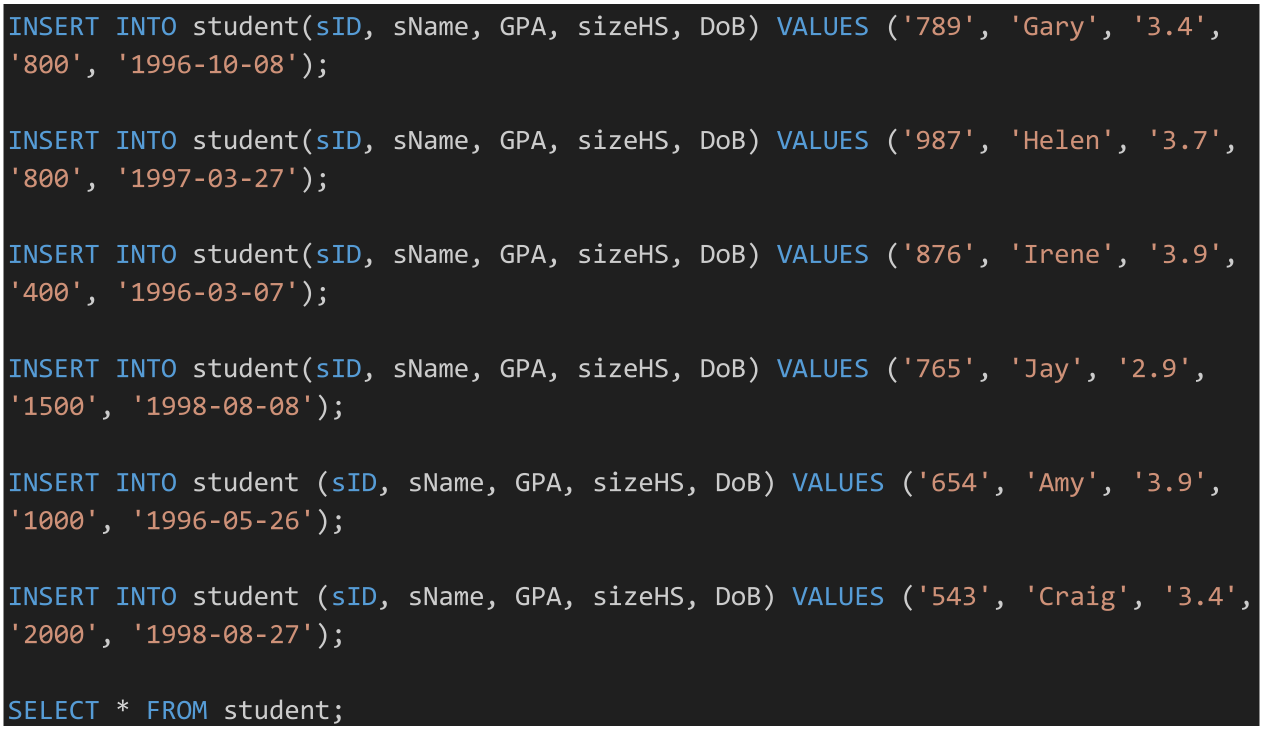
PRACTICAL ASSIGNMENT- 6

- Name: Hardik Arora
- Branch : Btech - CS
- Program: AIML
- University Roll No. : 2215500071
- Section: 2AC
- Class Roll No. : 28

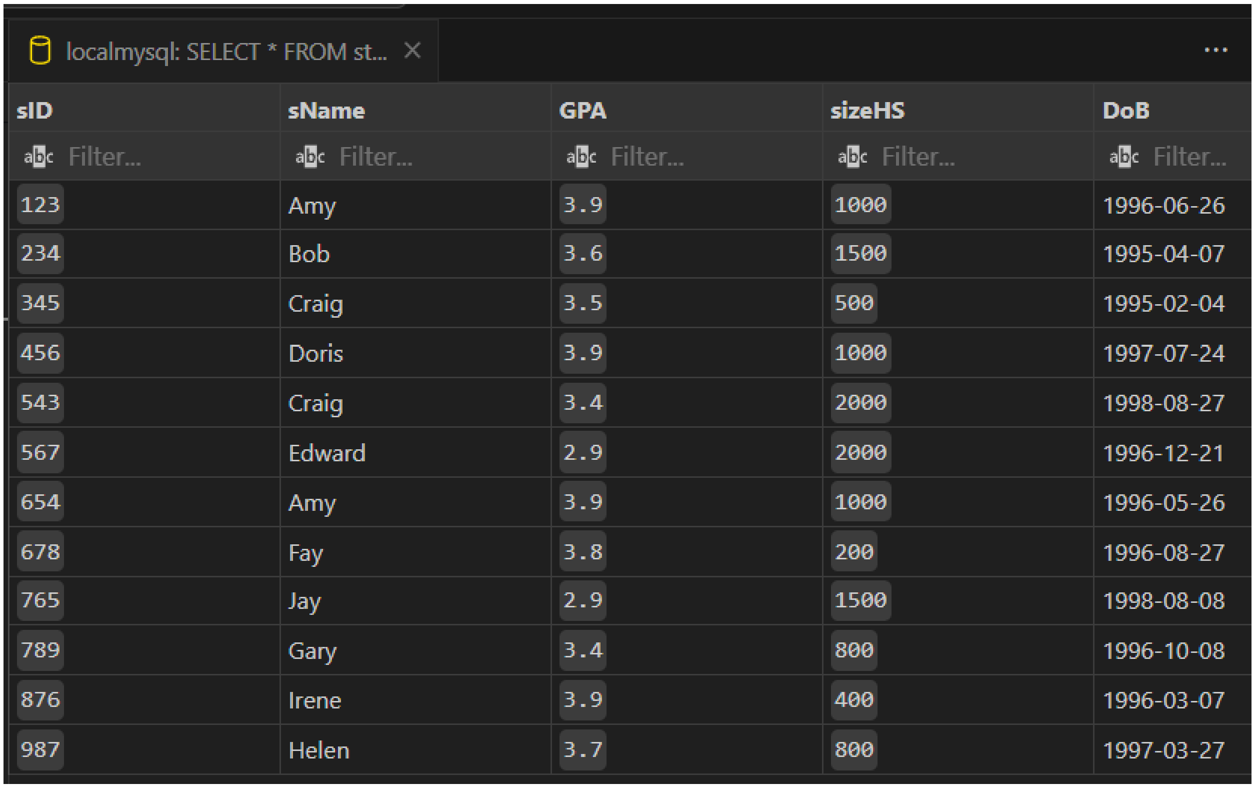
```sql
CREATE DATABASE P6;
USE P6;
```

```sql
CREATE TABLE IF NOT EXISTS Student (
    sID INT PRIMARY KEY,
    sName VARCHAR(50),
    GPA FLOAT,
    sizeHS INT NOT NULL,
    DoB VARCHAR(50)
);
```

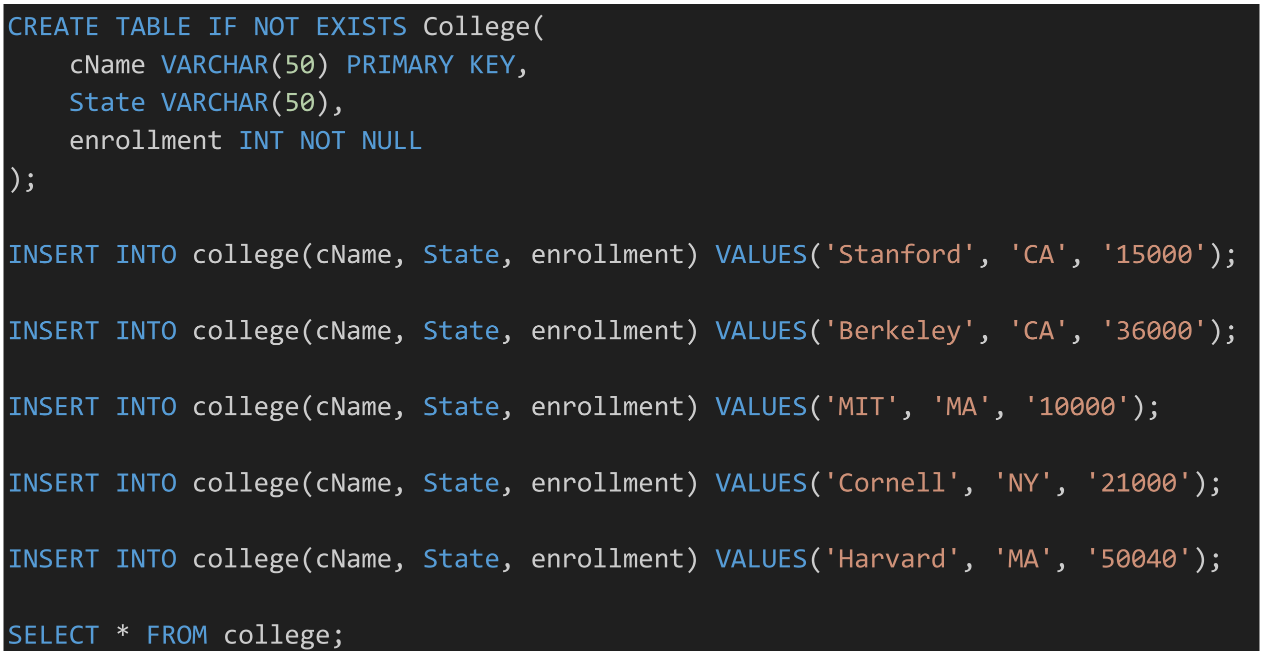
```sql
INSERT INTO student(sID, sName, GPA, sizeHS, DoB) VALUES ('123', 'Amy', '3.9',
'1000', '1996-06-26');

INSERT INTO student(sID, sName, GPA, sizeHS, DoB) VALUES ('234', 'Bob', '3.6',
'1500', '1995-04-07');

INSERT INTO student(sID, sName, GPA, sizeHS, DoB) VALUES ('345', 'Craig', '3.5',
'500', '1995-02-04');

INSERT INTO student(sID, sName, GPA, sizeHS, DoB) VALUES ('456', 'Doris', '3.9',
'1000', '1997-07-24');

INSERT INTO student(sID, sName, GPA, sizeHS, DoB) VALUES ('567', 'Edward', '2.9',
'2000', '1996-12-21');

INSERT INTO student(sID, sName, GPA, sizeHS, DoB) VALUES ('678', 'Fay', '3.8',
'200', '1996-08-27');
```



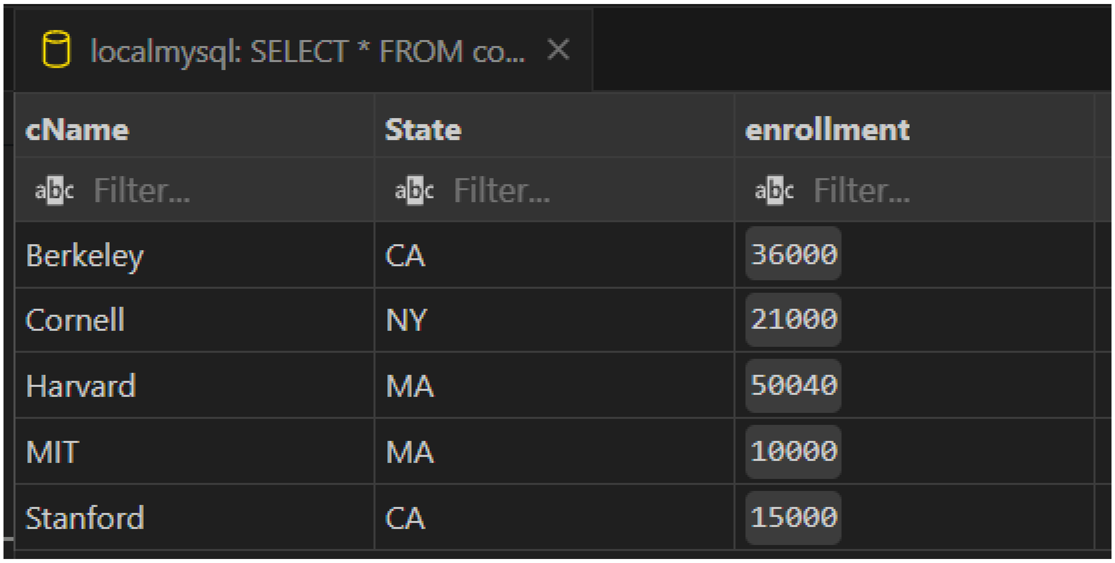
```
INSERT INTO student(sID, sName, GPA, sizeHS, DoB) VALUES ('789', 'Gary', '3.4', '800', '1996-10-08');

INSERT INTO student(sID, sName, GPA, sizeHS, DoB) VALUES ('987', 'Helen', '3.7', '800', '1997-03-27');

INSERT INTO student(sID, sName, GPA, sizeHS, DoB) VALUES ('876', 'Irene', '3.9', '400', '1996-03-07');

INSERT INTO student(sID, sName, GPA, sizeHS, DoB) VALUES ('765', 'Jay', '2.9', '1500', '1998-08-08');

INSERT INTO student (sID, sName, GPA, sizeHS, DoB) VALUES ('654', 'Amy', '3.9', '1000', '1996-05-26');

INSERT INTO student (sID, sName, GPA, sizeHS, DoB) VALUES ('543', 'Craig', '3.4', '2000', '1998-08-27');

SELECT * FROM student;
```



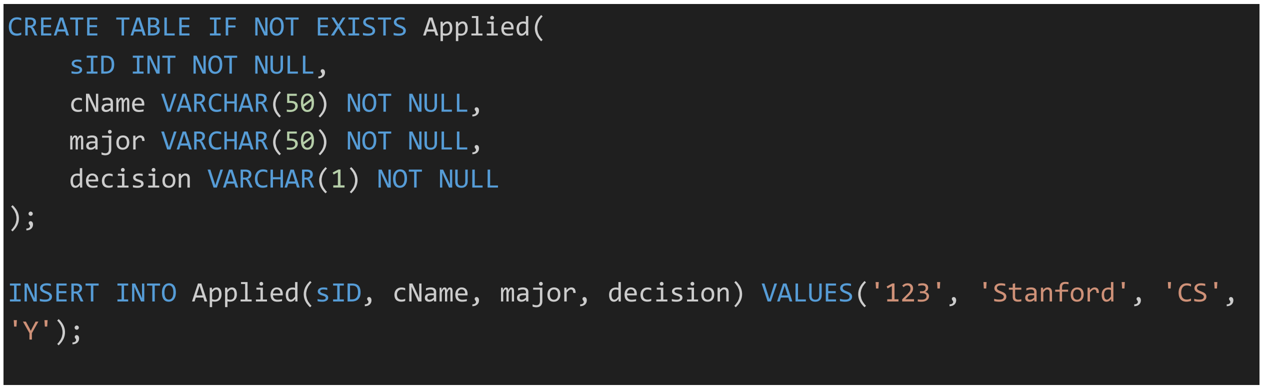
sID	sName	GPA	sizeHS	DoB
Filter...	Filter...	Filter...	Filter...	Filter...
123	Amy	3.9	1000	1996-06-26
234	Bob	3.6	1500	1995-04-07
345	Craig	3.5	500	1995-02-04
456	Doris	3.9	1000	1997-07-24
543	Craig	3.4	2000	1998-08-27
567	Edward	2.9	2000	1996-12-21
654	Amy	3.9	1000	1996-05-26
678	Fay	3.8	200	1996-08-27
765	Jay	2.9	1500	1998-08-08
789	Gary	3.4	800	1996-10-08
876	Irene	3.9	400	1996-03-07
987	Helen	3.7	800	1997-03-27



```sql
CREATE TABLE IF NOT EXISTS College(
    cName VARCHAR(50) PRIMARY KEY,
    State VARCHAR(50),
    enrollment INT NOT NULL
);

INSERT INTO college(cName, State, enrollment) VALUES('Stanford', 'CA', '15000');

INSERT INTO college(cName, State, enrollment) VALUES('Berkeley', 'CA', '36000');

INSERT INTO college(cName, State, enrollment) VALUES('MIT', 'MA', '10000');

INSERT INTO college(cName, State, enrollment) VALUES('Cornell', 'NY', '21000');

INSERT INTO college(cName, State, enrollment) VALUES('Harvard', 'MA', '50040');

SELECT * FROM college;
```



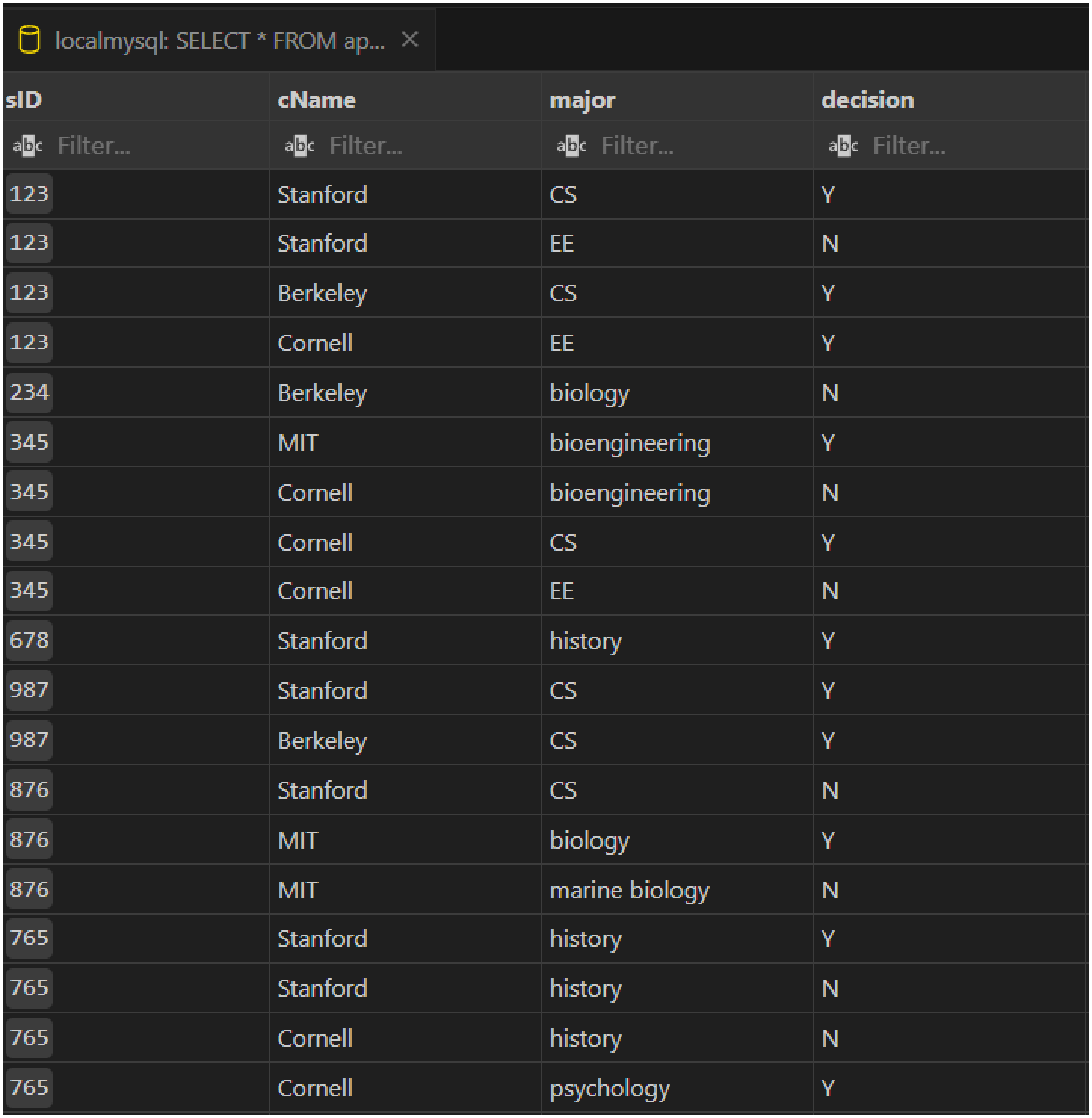
cName	State	enrollment
Filter...	Filter...	Filter...
Berkeley	CA	36000
Cornell	NY	21000
Harvard	MA	50040
MIT	MA	10000
Stanford	CA	15000



```sql
CREATE TABLE IF NOT EXISTS Applied(
    sID INT NOT NULL,
    cName VARCHAR(50) NOT NULL,
    major VARCHAR(50) NOT NULL,
    decision VARCHAR(1) NOT NULL
);

INSERT INTO Applied(sID, cName, major, decision) VALUES('123', 'Stanford', 'CS',
'Y');
```

```sql
INSERT INTO Applied(sID, cName, major, decision) VALUES('123', 'Stanford', 'EE',
'N');

INSERT INTO Applied(sID, cName, major, decision) VALUES('123', 'Berkeley', 'CS',
'Y');

INSERT INTO Applied(sID, cName, major, decision) VALUES('123', 'Cornell', 'EE',
'Y');

INSERT INTO Applied(sID, cName, major, decision) VALUES('234', 'Berkeley',
'biology', 'N');

INSERT INTO Applied(sID, cName, major, decision) VALUES('345', 'MIT',
'bioengineering', 'Y');

INSERT INTO Applied(sID, cName, major, decision) VALUES('345', 'Cornell',
'bioengineering', 'N');

INSERT INTO Applied(sID, cName, major, decision) VALUES('345', 'Cornell', 'CS',
'Y');

INSERT INTO Applied(sID, cName, major, decision) VALUES('345', 'Cornell', 'EE',
'N');

INSERT INTO Applied(sID, cName, major, decision) VALUES('678', 'Stanford',
'history', 'Y');

INSERT INTO Applied(sID, cName, major, decision) VALUES('987', 'Stanford', 'CS',
'Y');

INSERT INTO Applied(sID, cName, major, decision) VALUES('987', 'Berkeley', 'CS',
'Y');

INSERT INTO Applied(sID, cName, major, decision) VALUES('876', 'Stanford', 'CS',
'N');

INSERT INTO Applied(sID, cName, major, decision) VALUES('876', 'MIT', 'biology',
'Y');

INSERT INTO applied(sID, cName, major, decision) VALUES('876', 'MIT', 'marine
biology', 'N');

INSERT INTO Applied(sID, cName, major, decision) VALUES('765', 'Stanford',
'history', 'Y');
```

```
INSERT INTO applied(sID, cName, major, decision) VALUES('765', 'Stanford',
'history', 'N');

INSERT INTO applied(sID, cName, major, decision) VALUES('765', 'Cornell',
'history', 'N');

INSERT INTO applied(sID, cName, major, decision) VALUES('765', 'Cornell',
'psychology', 'Y');

INSERT INTO applied(sID, cName, major, decision) VALUES('543', 'MIT', 'CS', 'N');

SELECT * FROM applied;
```



sID	cName	major	decision
123	Stanford	CS	Y
123	Stanford	EE	N
123	Berkeley	CS	Y
123	Cornell	EE	Y
234	Berkeley	biology	N
345	MIT	bioengineering	Y
345	Cornell	bioengineering	N
345	Cornell	CS	Y
345	Cornell	EE	N
678	Stanford	history	Y
987	Stanford	CS	Y
987	Berkeley	CS	Y
876	Stanford	CS	N
876	MIT	biology	Y
876	MIT	marine biology	N
765	Stanford	history	Y
765	Stanford	history	N
765	Cornell	history	N
765	Cornell	psychology	Y

Write SQL queries for the following:

Q1. Create a new column DoB in Student table. (Datatype will be date)

```
ALTER TABLE Student
ADD COLUMN DoB DATE;
```

Q2. Insert DoB for each Student in corresponding table using above instance of Student table.

```sql
UPDATE Student SET DoB = '1996-06-26' WHERE sID = '123';
UPDATE Student SET DoB = '1995-04-07' WHERE sID = '234';
UPDATE Student SET DoB = '1995-02-04' WHERE sID = '345';
UPDATE Student SET DoB = '1997-07-24' WHERE sID = '456';
UPDATE Student SET DoB = '1996-12-21' WHERE sID = '567';
UPDATE Student SET DoB = '1996-08-27' WHERE sID = '678';
UPDATE Student SET DoB = '1996-10-08' WHERE sID = '789';
UPDATE Student SET DoB = '1997-03-27' WHERE sID = '987';
UPDATE Student SET DoB = '1996-03-07' WHERE sID = '876';
UPDATE Student SET DoB = '1998-08-08' WHERE sID = '765';
UPDATE Student SET DoB = '1996-05-26' WHERE sID = '654';
UPDATE Student SET DoB = '1998-08-27' WHERE sID = '543';
```

Q3. Find average of GPA round off to 2 decimal places.

```sql
SELECT ROUND(AVG(GPA), 2) AS average_gpa FROM Student;
```

Q4. Find year of DoB of Student having less than 1000.

```sql
SELECT YEAR(DoB) AS birth_year FROM Student WHERE sizeHS < 1000;
```

Q5. Compute Age of each student. (Hint: take difference between year of sysdate and Student’s DoB)

```sql
SELECT sName, YEAR(CURRENT_DATE()) - YEAR(DoB) - (DATE_FORMAT(CURRENT_DATE(),
'%m%d') < DATE_FORMAT(DoB, '%m%d')) AS age FROM Student;
```

Q6. Display name of all Students in uppercase and name of college they applied in lower case.

```sql
SELECT UPPER(sName) AS student_name, LOWER(cName) AS college_name FROM Student
JOIN Applied ON Student.sID = Applied.sID;
```

Q7. Find fourth alphabet of each student. (Hint: use substring)

```sql
SELECT SUBSTRING(sName, 4, 1) AS fourth_alphabet FROM Student;
```

Q8. Find sID and sName of student whose sName has string length greater than 3.

```sql
SELECT sID, sName FROM Student WHERE LENGTH(sName) > 3;
```

Q9. Find floor, ceiling and truncate (to one decimal place) value of average GPA.

```
SELECT FLOOR(AVG(GPA)) AS floor_value, CEILING(AVG(GPA)) AS ceiling_value,
TRUNCATE(AVG(GPA), 1) AS truncated_value FROM Student;
```

Q10. Display details of all students whose sID is even.

```
SELECT * FROM Student WHERE sID % 2 = 0;
```

Q11. Compute Square Root of 900 and 24^7.

```
SELECT SQRT(900) AS square_root_900, SQRT(247) AS square_root_247;
```

Q12. Consider the string “Peter Piper picked a peck of pickled peppers. A peck of pickled peppers Peter Piper picked. If Peter Piper picked a peck of pickled peppers, Where the peck of pickled peppers Peter Piper picked?” Find 6th occurrence of string ‘pick’. (Hint: use INSTR)

```
SELECT INSTR('Peter Piper picked a peck of pickled peppers. A peck of pickled
peppers Peter Piper picked. If Peter Piper picked a peck of pickled peppers,
Where the peck of pickled peppers Peter Piper picked?', 'pick', 1, 6) AS
sixth_occurrence_position;
```

Q13. Consider String ‘Satya Nadella’ replace this using the key (Hint: use translate)

```
SELECT TRANSLATE('Satya Nadella', 'aelNST123456y7', 'eaelNS1t234567y7') AS
replaced_string;
```

Q14. Display sID, sname and DoB in this format ‘February 26, 2014’

```
SELECT sID, sname, DATE_FORMAT(DoB, '%M %e, %Y') AS formatted_date FROM Student;
```

Q15. Convert the text ’26/02/2014’ to date.

```
SELECT STR_TO_DATE('26/02/2014', '%d/%m/%Y') AS converted_date;
```

Q16. Compute on which date is next Saturday and last day of this month?

```
SELECT DATE_ADD(CURRENT_DATE(), INTERVAL (7 - DAYOFWEEK(CURRENT_DATE())) DAY) AS
next_saturday;
SELECT LAST_DAY(CURRENT_DATE()) AS last_day_of_month;
```

Exercise:

Q1. Display sID, sname and DoB in this format ‘26th Feburary, 2014’

```
SELECT sID, sname, DATE_FORMAT(DoB, '%D %M, %Y') AS formatted_date FROM Student;
```

Q2.   Display sID, sname and DoB in this format ‘26/02/2014’

```
SELECT sID, sname, DATE_FORMAT(DoB, '%d/%m/%Y') AS formatted_date FROM Student;
```

Q3. Add 5 months to DoB of Edward?

```
UPDATE Student SET DoB = DATE_ADD(DoB, INTERVAL 5 MONTH) WHERE sName = 'Edward';
```

Q4. Display last day of DoB of Amy?

```
SELECT LAST_DAY(DoB) AS last_day_of_birth FROM Student WHERE sName = 'Amy';
```

Q5. Display next Sunday of DoB of Doris?

```
SELECT DATE_ADD(DoB, INTERVAL (8 - DAYOFWEEK(DoB)) DAY) AS next_sunday FROM
Student WHERE sName = 'Doris';
```

- Name: Hardik Arora
- Branch :   Btech - CS
- Program: AIML
- University Roll No. :   2215500071
- Section: 2AC
- Class Roll No. :   28
- Submitted to: Ayushi Mam
- Submitted on: 06-05-2024