PRACTICAL ASSIGNMENT- 8

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

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

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

```
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;
```

localmysql: SELECT * FROM st...

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;
```

localmysql: SELECT * FROM co...

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');
```

```
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');
```

```sql
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;
```

localmysql: SELECT * FROM ap...

sID	cName	major	decision
Filter...	Filter...	Filter...	Filter...
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. As we need to notify in the system the birthday of each student, kindly
create an index DoBindex on the column DoB of the Student table.

CREATE INDEX DoBindex ON Student(DoB);
```

```
- Q2. Which index would be more suitable for the major in Apply? Create a Bitmap Index named MAJORindex.
CREATE BITMAP INDEX MAJORindex ON Apply(major);
```

```
-- Q3. Remove the index on the DoB column.

DROP INDEX DoBindex;
```

```
-- Q4. Create a Unique index ENRindex on enrollment.

CREATE UNIQUE INDEX ENRindex ON College(enrollment);
```

```
-- Q5. Create a composite Unique index SCMindex on Apply using columns sID, cName, major.

CREATE UNIQUE INDEX SCMindex ON Apply(sID, cName, major);
```

```
-- Q6. Create a composite index on Apply using columns cName and major. Name this index as CMapplyINDX.

CREATE INDEX CMapplyINDX ON Apply(cName, major);
```

```
-- Q7. Create sequence sID_seq with the following parameters: increment by 3, cycle, cache 4, and which will generate the numbers among 988 to 999.

CREATE SEQUENCE sID_seq
INCREMENT BY 3
START WITH 988
MAXVALUE 999
CYCLE
CACHE 4;
```

```
-- Q8. Display the next value of Sequence sID_seq.

SELECT sID_seq.nextval FROM dual;
```

```
-- Q9. A new student entered the database named Eric with the next sID from sequence sID_seq having GPA 9.9, sizeHS 9999, DoB as '23-Apr-98' to the Student table.
```

```sql
INSERT INTO Student (sID, sName, GPA, sizeHS, DoB)
VALUES (sID_seq.nextval, 'Eric', 9.9, 9999, TO_DATE('23-Apr-98', 'DD-MON-YY'));
```

```sql
-- Q10. Now, another boy registered in our system named Troy with the next sID
from sequence sID_seq having GPA 9.8 and sizeHS 989 and Dob as '25-Nov-99'.

INSERT INTO Student (sID, sName, GPA, sizeHS, DoB)
VALUES (sID_seq.nextval, 'Troy', 9.8, 989, TO_DATE('25-Nov-99', 'DD-MON-YY'));
```

```sql
-- Q11. Display details of the Student table and observe the newly inserted Eric
and Troy sID.

SELECT * FROM Student;
```

```sql
-- Q12. Create a sequence GPA_seq having a maximum value of 5 and a minimum value
of 3. You are supposed to start the sequence with 5 and decrement the sequence
with -1, cycle and no cache.

CREATE SEQUENCE GPA_seq
START WITH 5
INCREMENT BY -1
MAXVALUE 5
MINVALUE 3
CYCLE
NOCACHE;
```

```sql
-- Q13. Update GPA of Eric to the next value of sequence GPA_seq.

UPDATE Student
SET GPA = GPA_seq.nextval
WHERE sName = 'Eric';
```

```sql
-- Q14. Insert student Jack with sID from sID_seq, GPA from GPA_seq, sizeHS as
1500, and DoB as '22-OCT-97'.

INSERT INTO Student (sID, sName, GPA, sizeHS, DoB)
VALUES (sID_seq.nextval, 'Jack', GPA_seq.nextval, 1500, TO_DATE('22-OCT-97', 'DD-
MON-YY'));
```

```sql
-- Q15. Display details of the Student Table and observe GPA and sID of Jack.

SELECT * FROM Student WHERE sName = 'Jack';
```

```sql
-- Q16. Display the next value of sequence GPA_seq.

SELECT GPA_seq.nextval FROM dual;
```

```sql
-- Q17. Drop sequence sID_seq.

DROP SEQUENCE sID_seq;
```

```sql
-- Q18. Drop sequence GPA_seq.

DROP SEQUENCE GPA_seq;
```

- 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