Lab Assignment 2(a)

Question:

Create the following tables:

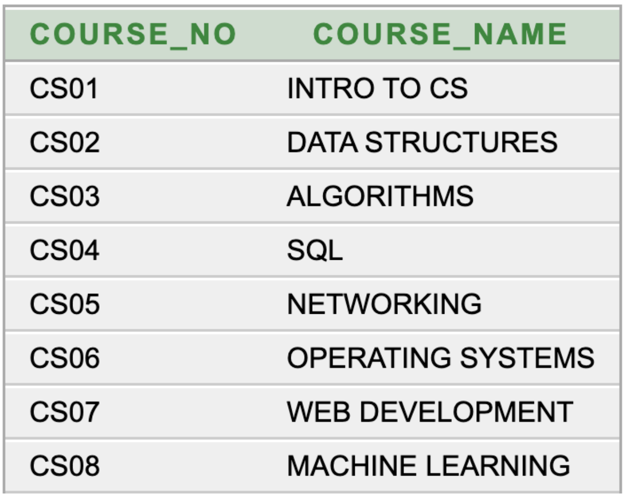
- Course(course_no char(4), course_name varchar(20))
- Course_fee(course_no char(4), full_part char(1) (F/P), fees number(10))
- course_no and full_part should be unique
- Student(prospectus_no number(10), name varchar(20), address varchar(30), phone_no number(11), D_O_B date, total_amt number(10,2), amt_paid number(10,2), installment char(1) (I/F))
- Installment(prospectus_no number(10) (foreign key) on delete cascade, installment_amt number(10,2), due_dt date, paid char(1) (P,U))
- prospectus_no and due_dt should be unique
- Course_taken(prospectus_no number(10) (foreign key), course_no char(4), start_dt date, full_part char(1) (F/P), time_slot char(2), performance varchar(20))

Answer:

```
CREATE TABLE SCOURSE (COURSE_NO CHAR(4) UNIQUE, COURSE_NAME VARCHAR(20));

INSERT INTO SCOURSE (COURSE_NO, COURSE_NAME) VALUES ('CS01', 'INTRO TO CS');
INSERT INTO SCOURSE (COURSE_NO, COURSE_NAME) VALUES ('CS02', 'DATA STRUCTURES');
INSERT INTO SCOURSE (COURSE_NO, COURSE_NAME) VALUES ('CS03', 'ALGORITHMS');
INSERT INTO SCOURSE (COURSE_NO, COURSE_NAME) VALUES ('CS04', 'DATABASES');
INSERT INTO SCOURSE (COURSE_NO, COURSE_NAME) VALUES ('CS05', 'NETWORKING');
INSERT INTO SCOURSE (COURSE_NO, COURSE_NAME) VALUES ('CS06', 'OPERATING SYSTEMS');
INSERT INTO SCOURSE (COURSE_NO, COURSE_NAME) VALUES ('CS07', 'WEB DEVELOPMENT');
INSERT INTO SCOURSE (COURSE_NO, COURSE_NAME) VALUES ('CS08', 'MACHINE LEARNING');

SELECT * FROM SCOURSE;
```

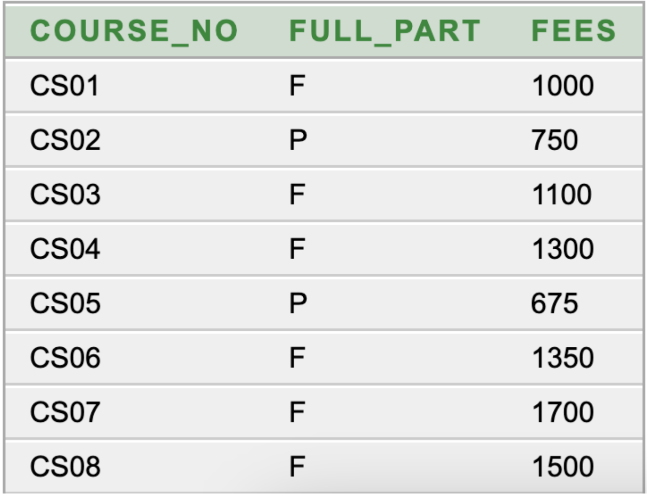


COURSE_NO	COURSE_NAME
CS01	INTRO TO CS
CS02	DATA STRUCTURES
CS03	ALGORITHMS
CS04	SQL
CS05	NETWORKING
CS06	OPERATING SYSTEMS
CS07	WEB DEVELOPMENT
CS08	MACHINE LEARNING

```
CREATE TABLE SCOURSE_FEE (COURSE_NO CHAR(4) UNIQUE, FULL_PART CHAR(1), FEES NUMBER(10), FOREIGN KEY(COURSE_NO) REFERENCES SCOURSE(COURSE_NO));

INSERT INTO SCOURSE_FEE (COURSE_NO, FULL_PART, FEES) VALUES ('CS01', 'F', 1000);
INSERT INTO SCOURSE_FEE (COURSE_NO, FULL_PART, FEES) VALUES ('CS02', 'P', 750);
INSERT INTO SCOURSE_FEE (COURSE_NO, FULL_PART, FEES) VALUES ('CS03', 'F', 1100);
INSERT INTO SCOURSE_FEE (COURSE_NO, FULL_PART, FEES) VALUES ('CS04', 'F', 1300);
INSERT INTO SCOURSE_FEE (COURSE_NO, FULL_PART, FEES) VALUES ('CS05', 'P', 675);
INSERT INTO SCOURSE_FEE (COURSE_NO, FULL_PART, FEES) VALUES ('CS06', 'F', 1350);
INSERT INTO SCOURSE_FEE (COURSE_NO, FULL_PART, FEES) VALUES ('CS07', 'F', 1700);
INSERT INTO SCOURSE_FEE (COURSE_NO, FULL_PART, FEES) VALUES ('CS08', 'F', 1500);
```

```
SELECT * FROM SCOURSE_FEE;
```

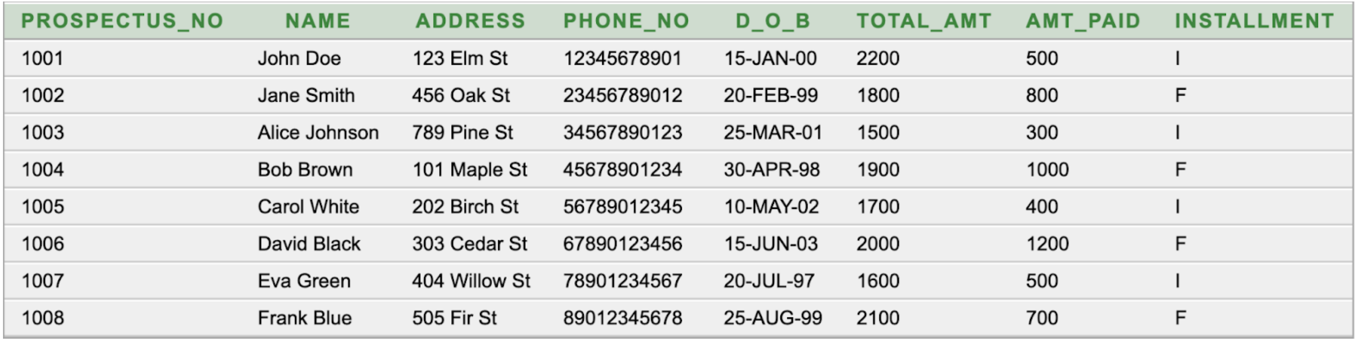


COURSE_NO	FULL_PART	FEES
CS01	F	1000
CS02	P	750
CS03	F	1100
CS04	F	1300
CS05	P	675
CS06	F	1350
CS07	F	1700
CS08	F	1500

```
CREATE TABLE SSSTUDENT (PROSPECTUS_NO NUMBER(10) UNIQUE, NAME
VARCHAR(20), ADDRESS VARCHAR(30), PHONE_NO NUMBER(11), D_O_B DATE,
TOTAL_AMT NUMBER(10, 2), AMT_PAID NUMBER(10, 2), INSTALLMENT CHAR(1));
```

```
INSERT INTO SSSTUDENT VALUES (1001, 'John Doe', '123 Elm St', 12345678901,
TO_DATE('2000-01-15', 'YYYY-MM-DD'), 2200.00, 500.00, 'I');
INSERT INTO SSSTUDENT VALUES (1002, 'Jane Smith', '456 Oak St', 23456789012,
TO_DATE('1999-02-20', 'YYYY-MM-DD'), 1800.00, 800.00, 'F');
INSERT INTO SSSTUDENT VALUES (1003, 'Alice Johnson', '789 Pine St', 34567890123,
TO_DATE('2001-03-25', 'YYYY-MM-DD'), 1500.00, 300.00, 'I');
INSERT INTO SSSTUDENT VALUES (1004, 'Bob Brown', '101 Maple St', 45678901234,
TO_DATE('1998-04-30', 'YYYY-MM-DD'), 1900.00, 1000.00, 'F');
INSERT INTO SSSTUDENT VALUES (1005, 'Carol White', '202 Birch St', 56789012345,
TO_DATE('2002-05-10', 'YYYY-MM-DD'), 1700.00, 400.00, 'I');
INSERT INTO SSSTUDENT VALUES (1006, 'David Black', '303 Cedar St', 67890123456,
TO_DATE('2003-06-15', 'YYYY-MM-DD'), 2000.00, 1200.00, 'F');
INSERT INTO SSSTUDENT VALUES (1007, 'Eva Green', '404 Willow St', 78901234567,
TO_DATE('1997-07-20', 'YYYY-MM-DD'), 1600.00, 500.00, 'I');
INSERT INTO SSSTUDENT VALUES (1008, 'Frank Blue', '505 Fir St', 89012345678,
TO_DATE('1999-08-25', 'YYYY-MM-DD'), 2100.00, 700.00, 'F');
```

```
SELECT * FROM SSSTUDENT;
```

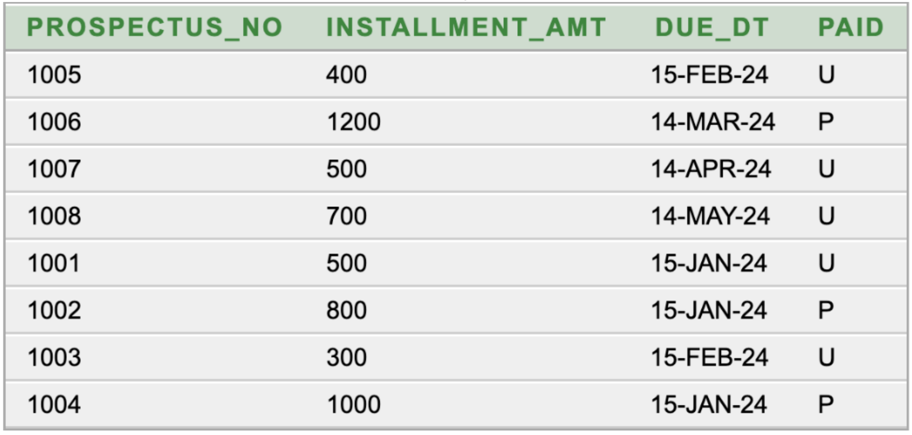


PROSPECTUS_NO	NAME	ADDRESS	PHONE_NO	D_O_B	TOTAL_AMT	AMT_PAID	INSTALLMENT
1001	John Doe	123 Elm St	12345678901	15-JAN-00	2200	500	I
1002	Jane Smith	456 Oak St	23456789012	20-FEB-99	1800	800	F
1003	Alice Johnson	789 Pine St	34567890123	25-MAR-01	1500	300	I
1004	Bob Brown	101 Maple St	45678901234	30-APR-98	1900	1000	F
1005	Carol White	202 Birch St	56789012345	10-MAY-02	1700	400	I
1006	David Black	303 Cedar St	67890123456	15-JUN-03	2000	1200	F
1007	Eva Green	404 Willow St	78901234567	20-JUL-97	1600	500	I
1008	Frank Blue	505 Fir St	89012345678	25-AUG-99	2100	700	F

```
CREATE TABLE SINSTALLMENT (PROSPECTUS_NO NUMBER(10) UNIQUE,
INSTALLMENT_AMT NUMBER(10,2), DUE_DT DATE UNIQUE, PAID CHAR(1), FOREIGN
KEY(PROSPECTUS_NO) REFERENCES SSSTUDENT(PROSPECTUS_NO));

INSERT INTO SINSTALLMENT VALUES (1001, 500.00, TO_DATE('2024-01-15', 'YYYY-MM-DD'),
'U');
INSERT INTO SINSTALLMENT VALUES (1002, 800.00, TO_DATE('2024-01-15', 'YYYY-MM-DD'),
'P');
INSERT INTO SINSTALLMENT VALUES (1003, 300.00, TO_DATE('2024-02-15', 'YYYY-MM-DD'),
'U');
INSERT INTO SINSTALLMENT VALUES (1004, 1000.00, TO_DATE('2024-01-15', 'YYYY-MM-
DD'), 'P');
INSERT INTO SINSTALLMENT VALUES (1005, 400.00, TO_DATE('2024-02-15', 'YYYY-MM-DD'),
'U');
INSERT INTO SINSTALLMENT VALUES (1006, 1200.00, TO_DATE('2024-03-14', 'YYYY-MM-
DD'), 'P');
INSERT INTO SINSTALLMENT VALUES (1007, 500.00, TO_DATE('2024-04-14', 'YYYY-MM-DD'),
'U');
INSERT INTO SINSTALLMENT VALUES (1008, 700.00, TO_DATE('2024-05-14', 'YYYY-MM-DD'),
'U');

SELECT * FROM SINSTALLMENT;
```



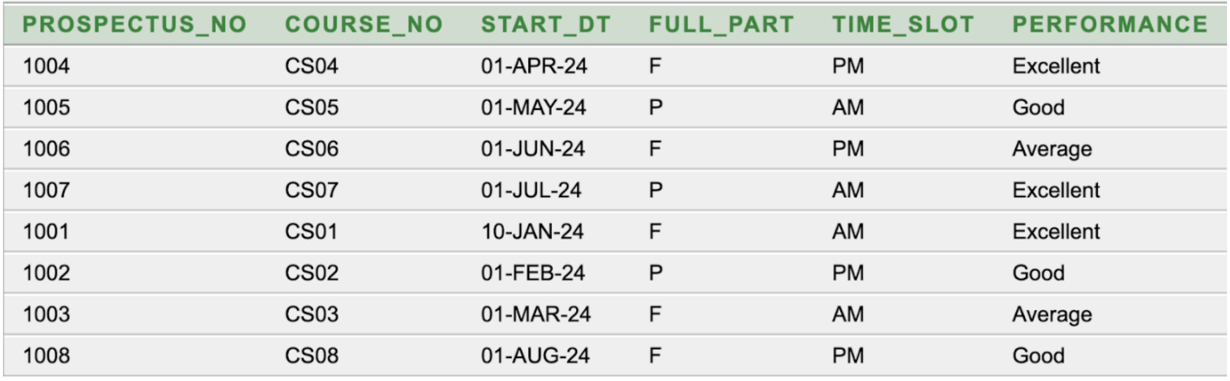
PROSPECTUS_NO	INSTALLMENT_AMT	DUE_DT	PAID
1005	400	15-FEB-24	U
1006	1200	14-MAR-24	P
1007	500	14-APR-24	U
1008	700	14-MAY-24	U
1001	500	15-JAN-24	U
1002	800	15-JAN-24	P
1003	300	15-FEB-24	U
1004	1000	15-JAN-24	P

```
CREATE TABLE SCOURSE_TAKEN (PROSPECTUS_NO NUMBER(10), COURSE_NO CHAR(4),
START_DT DATE, FULL_PART CHAR(1), TIME_SLOT CHAR(2), PERFORMANCE
VARCHAR(20), FOREIGN KEY(PROSPECTUS_NO) REFERENCES
SSSTUDENT(PROSPECTUS_NO), FOREIGN KEY(COURSE_NO) REFERENCES
SCOURSE(COURSE_NO));

INSERT INTO SCOURSE_TAKEN VALUES (1001, 'CS01', TO_DATE('2024-01-10', 'YYYY-MM-
DD'), 'F', 'AM', 'Excellent');
INSERT INTO SCOURSE_TAKEN VALUES (1002, 'CS02', TO_DATE('2024-02-01', 'YYYY-MM-
DD'), 'P', 'PM', 'Good');
INSERT INTO SCOURSE_TAKEN VALUES (1003, 'CS03', TO_DATE('2024-03-01', 'YYYY-MM-
DD'), 'F', 'AM', 'Average');
INSERT INTO SCOURSE_TAKEN VALUES (1004, 'CS04', TO_DATE('2024-04-01', 'YYYY-MM-
DD'), 'F', 'PM', 'Excellent');
INSERT INTO SCOURSE_TAKEN VALUES (1005, 'CS05', TO_DATE('2024-05-01', 'YYYY-MM-
DD'), 'P', 'AM', 'Good');
INSERT INTO SCOURSE_TAKEN VALUES (1006, 'CS06', TO_DATE('2024-06-01', 'YYYY-MM-
DD'), 'F', 'PM', 'Average');
INSERT INTO SCOURSE_TAKEN VALUES (1007, 'CS07', TO_DATE('2024-07-01', 'YYYY-MM-
DD'), 'P', 'AM', 'Excellent');
```

INSERT INTO SCOURSE_TAKEN VALUES (1008, 'CS08', TO_DATE('2024-08-01', 'YYYY-MM-DD'), 'F', 'PM', 'Good');

SELECT * FROM SCOURSE_TAKEN;

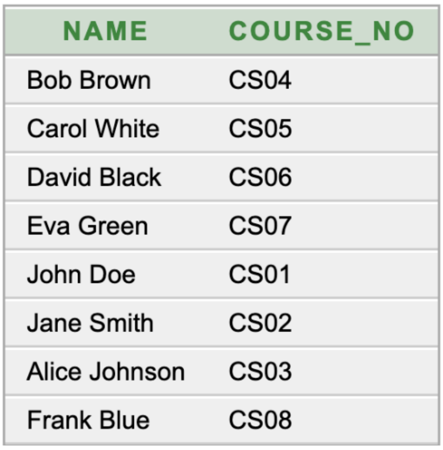


PROSPECTUS_NO	COURSE_NO	START_DT	FULL_PART	TIME_SLOT	PERFORMANCE
1004	CS04	01-APR-24	F	PM	Excellent
1005	CS05	01-MAY-24	P	AM	Good
1006	CS06	01-JUN-24	F	PM	Average
1007	CS07	01-JUL-24	P	AM	Excellent
1001	CS01	10-JAN-24	F	AM	Excellent
1002	CS02	01-FEB-24	P	PM	Good
1003	CS03	01-MAR-24	F	AM	Average
1008	CS08	01-AUG-24	F	PM	Good

Question:

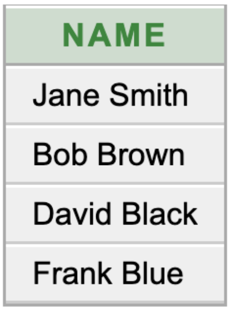
Perform the following queries on the tables just created :

1. Retrieve name and course no of all the students.
   - ➢ SELECT S.NAME, CT.COURSE_NO FROM SSSTUDENT S JOIN SCOURSE_TAKEN CT ON S.PROSPECTUS_NO = CT.PROSPECTUS_NO;



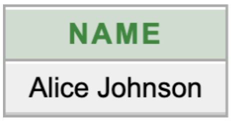
NAME	COURSE_NO
Bob Brown	CS04
Carol White	CS05
David Black	CS06
Eva Green	CS07
John Doe	CS01
Jane Smith	CS02
Alice Johnson	CS03
Frank Blue	CS08

2. List the names of students who have paid the full amount at the time of admission.
   - ➢ SELECT NAME FROM SSSTUDENT WHERE INSTALLMENT = 'F';



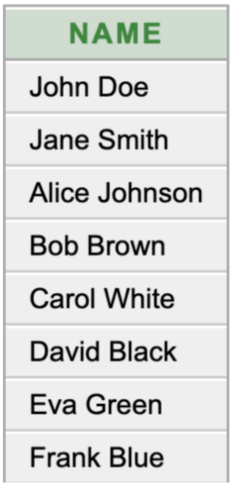
NAME
Jane Smith
Bob Brown
David Black
Frank Blue

3. Find the names of students starting with A.
   - ➢ SELECT NAME FROM SSSTUDENT WHERE NAME LIKE 'A%';



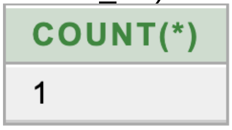
NAME
Alice Johnson

4. Print the names of students whose total amount is not equal to amount due.
   - ➢ SELECT NAME FROM SSSTUDENT WHERE TOTAL_AMT <> AMT_PAID;



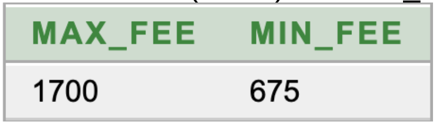
NAME
John Doe
Jane Smith
Alice Johnson
Bob Brown
Carol White
David Black
Eva Green
Frank Blue

5. Count the number of students who have joined in current year, current month.
   - ➢ SELECT COUNT(*) FROM SCOURSE_TAKEN WHERE EXTRACT(YEAR FROM START_DT) = EXTRACT(YEAR FROM SYSDATE) AND EXTRACT(MONTH FROM START_DT) = EXTRACT(MONTH FROM SYSDATE);



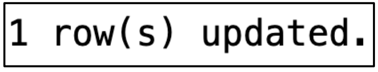
COUNT(*)
1

6. Determine the maximum and minimum course fees.
   - ➢ SELECT MAX(FEES) AS MAX_FEE, MIN(FEES) AS MIN_FEE FROM SCOURSE_FEE;



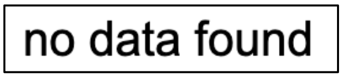
MAX_FEE	MIN_FEE
1700	675

7. Increase the fee of databases by 50%.
   - ➢ UPDATE SCOURSE_FEE SET FEES = FEES * 1.50 WHERE COURSE_NO = 'CS04' AND FULL_PART = 'F';



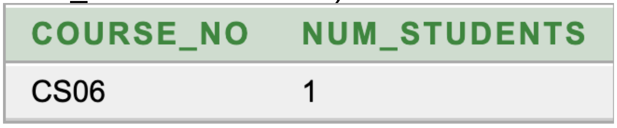
```
1 row(s) updated.
```

8. Print the details of courses whose fees are between 5000 and 10000.
   - ➢ SELECT * FROM SCOURSE_FEE WHERE FEES BETWEEN 5000 AND 10000;



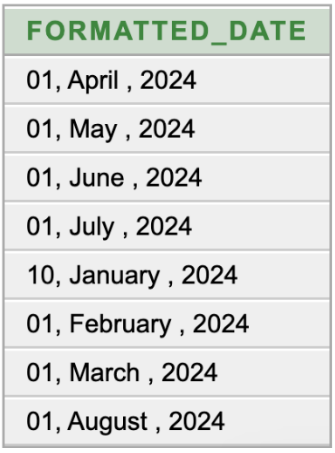
```
no data found
```

9. Find out in which course maximum number of students have taken admission.
   - ➢ SELECT COURSE_NO, NUM_STUDENTS FROM (SELECT COURSE_NO, COUNT(*) AS NUM_STUDENTS FROM SCOURSE_TAKEN GROUP BY COURSE_NO ORDER BY NUM_STUDENTS DESC) WHERE ROWNUM = 1;



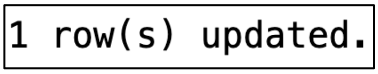
COURSE_NO	NUM_STUDENTS
CS06	1

10. Display the admission date in Date, Month, Year format.
    - ➢ SELECT TO_CHAR(START_DT, 'DD, Month, YYYY') AS FORMATTED_DATE FROM SCOURSE_TAKEN;



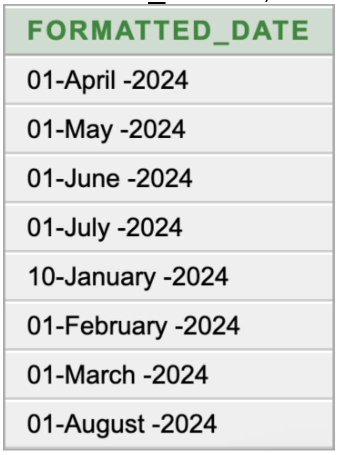
FORMATTED_DATE
01, April , 2024
01, May , 2024
01, June , 2024
01, July , 2024
10, January , 2024
01, February , 2024
01, March , 2024
01, August , 2024

11. Change the course_name from Databases to SQL.
   - ➢ UPDATE SCOURSE SET COURSE_NAME = 'SQL' WHERE COURSE_NAME = 'DATABASES';



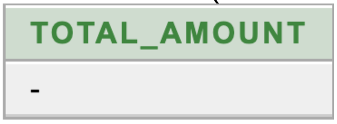
```
1 row(s) updated.
```

12. Display the admission date in DD-MONTH-YYYY format.
   - ➢ SELECT TO_CHAR(START_DT, 'DD-Month-YYYY') AS FORMATTED_DATE FROM SCOURSE_TAKEN;



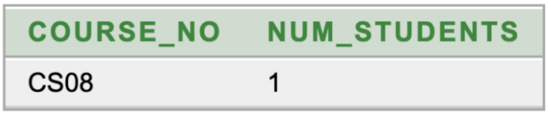
FORMATTED_DATE
01-April -2024
01-May -2024
01-June -2024
01-July -2024
10-January -2024
01-February -2024
01-March -2024
01-August -2024

13. Get the sum of amount to be collected from students in this month.
   - ➢ SELECT SUM(INSTALLMENT_AMT) AS TOTAL_AMOUNT FROM SINSTALLMENT WHERE EXTRACT(MONTH FROM DUE_DT) = EXTRACT(MONTH FROM SYSDATE) AND EXTRACT(YEAR FROM DUE_DT) = EXTRACT(YEAR FROM SYSDATE);



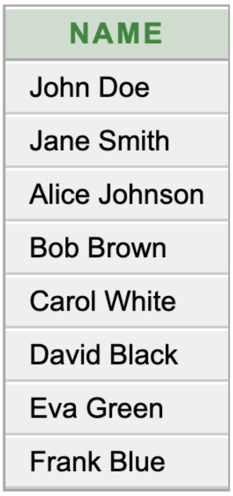
TOTAL_AMOUNT
-

14. Find out in which course the maximum number of students have taken admission in the current month.
   - ➢ SELECT COURSE_NO, NUM_STUDENTS FROM (SELECT COURSE_NO, COUNT(*) AS NUM_STUDENTS FROM SCOURSE_TAKEN WHERE EXTRACT(MONTH FROM START_DT) = EXTRACT(MONTH FROM SYSDATE) AND EXTRACT(YEAR FROM START_DT) = EXTRACT(YEAR FROM SYSDATE) GROUP BY COURSE_NO ORDER BY NUM_STUDENTS DESC) WHERE ROWNUM = 1;



COURSE_NO	NUM_STUDENTS
CS08	1

15. Select the students who have not yet paid full amount of fees.
    - ➢ SELECT NAME FROM SSSTUDENT WHERE TOTAL_AMT > AMT_PAID;



NAME
John Doe
Jane Smith
Alice Johnson
Bob Brown
Carol White
David Black
Eva Green
Frank Blue