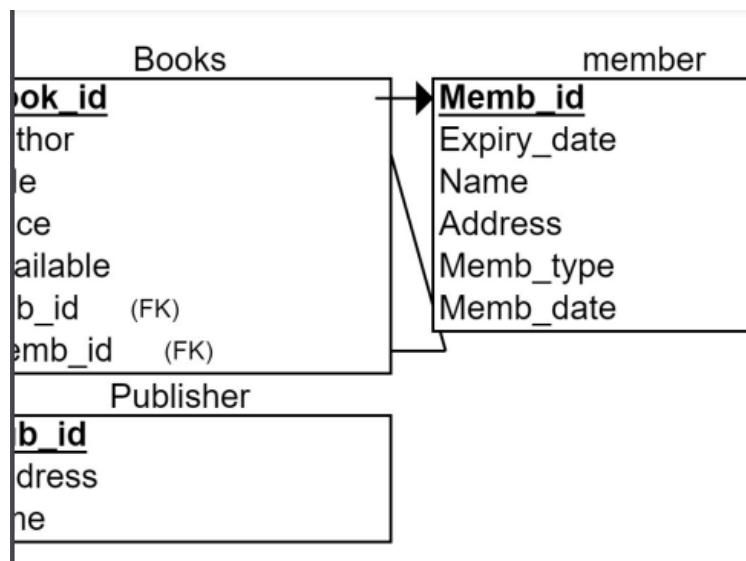
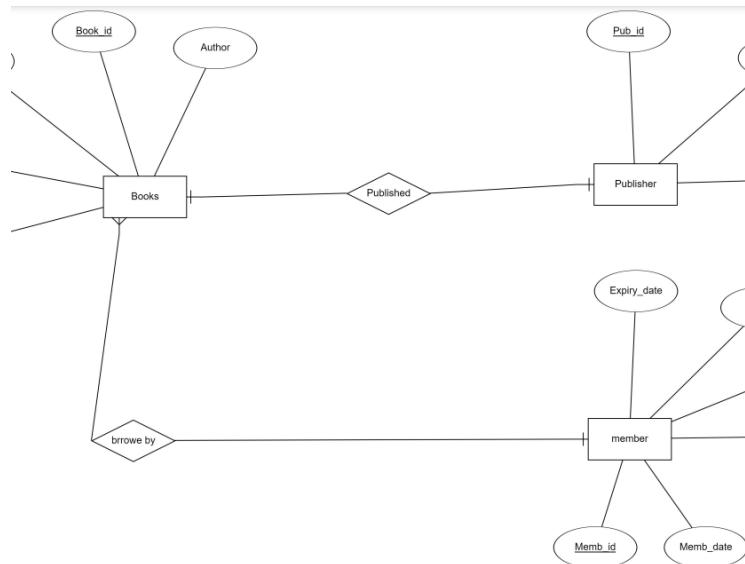
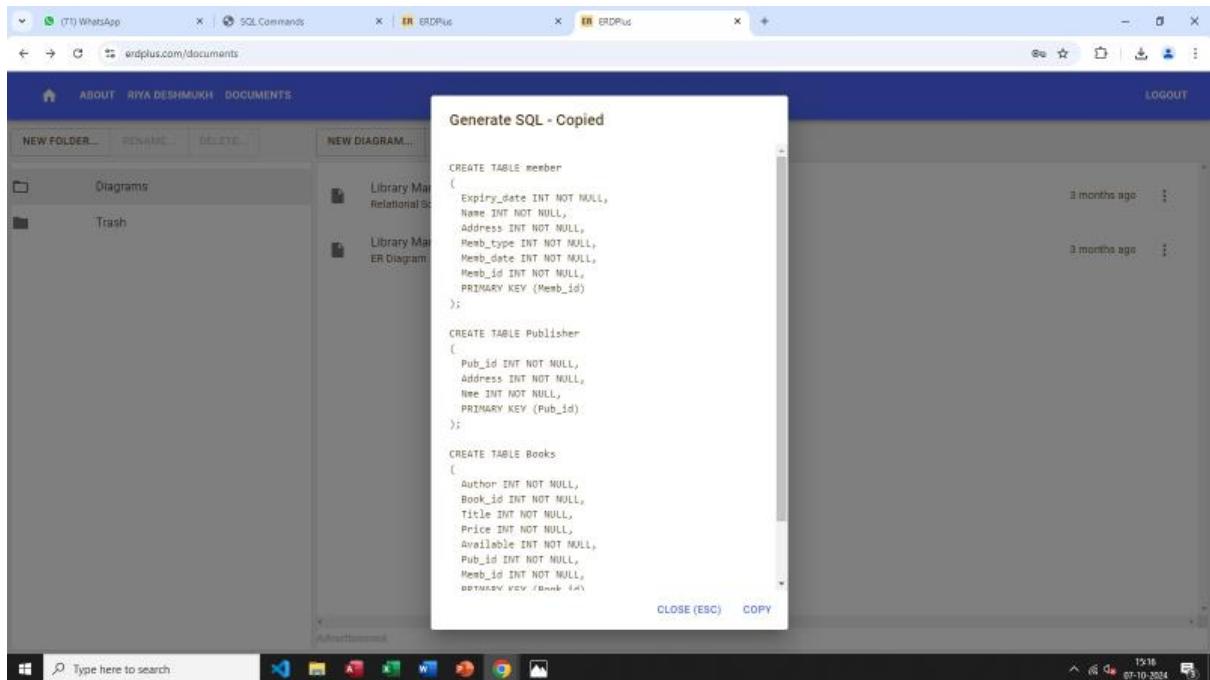


Assignment 1

Output:-





Assignment 8

Input:-

```
CREATE TABLE T3
```

```
(  
    Roll_no  NUMBER(10) NOT NULL,  
    name     VARCHAR2(50),  
    issuedate DATE,  
    book     VARCHAR2(10),  
    status   CHAR(1),  
    CONSTRAINT pk1_T3 PRIMARY KEY (Roll_no)  
);
```

```
CREATE TABLE T4
```

```
(  
    Roll_no  NUMBER(10) NOT NULL,  
    name     VARCHAR2(50),  
    issuedate DATE,  
    book     VARCHAR2(10),  
    status   CHAR(1),  
    ts       TIMESTAMP  
);
```

```
INSERT ALL
```

```
    INTO T3 VALUES (101, 'Sita', TO_DATE('22-Dec-2022', 'DD-MON-YYYY'), 'TOC', 'I')
```

```
    INTO T3 VALUES (102, 'Mita', TO_DATE('20-Sep-2023', 'DD-MON-YYYY'), 'TOC', 'I')
```

```
    INTO T3 VALUES (103, 'Gita', TO_DATE('12-Nov-2023', 'DD-MON-YYYY'), 'TOC', 'I')
```

```
SELECT * FROM dual;
```

```
SELECT * FROM T3;
```

```
TRUNCATE TABLE T3;
```

```
TRUNCATE TABLE T4;
```

```

CREATE OR REPLACE TRIGGER lib_audit
AFTER DELETE OR INSERT OR UPDATE ON T3
FOR EACH ROW
BEGIN
  IF INSERTING THEN
    INSERT INTO T4 (roll_no, name, issuedate, book, status, ts)
    VALUES (:new.roll_no, :new.name, :new.issuedate, :new.book, :new.status, SYSTIMESTAMP);
  END IF;

  IF UPDATING THEN
    INSERT INTO T4 (roll_no, name, issuedate, book, status, ts)
    VALUES (:old.roll_no, :old.name, :old.issuedate, :old.book, :old.status, SYSTIMESTAMP);
    INSERT INTO T4 (roll_no, name, issuedate, book, status, ts)
    VALUES (:new.roll_no, :new.name, :new.issuedate, :new.book, :new.status, SYSTIMESTAMP);
  END IF;

  IF DELETING THEN
    INSERT INTO T4 (roll_no, name, issuedate, book, status, ts)
    VALUES (:old.roll_no, :old.name, :old.issuedate, :old.book, :old.status, SYSTIMESTAMP);
  END IF;
END;
/
INSERT INTO T3 VALUES (104, 'ABCD', TO_DATE('12-AUG-2024', 'DD-MON-YYYY'), 'AI', 'I');
SELECT * FROM T3;
SELECT * FROM T4;
UPDATE T3 SET name='Nita' WHERE roll_no=104;
SELECT * FROM T4;
DELETE FROM T3 WHERE roll_no=104;
SELECT * FROM T3;
SELECT * FROM T4;

```

Output:-

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
CREATE TABLE T3
(
    Roll_no NUMBER(10) NOT NULL,
    name VARCHAR2(50),
    IssueDate DATE,
    book VARCHAR2(10),
    status CHAR(1),
    CONSTRAINT pk1_T3 PRIMARY KEY (Roll_no)
);

CREATE TABLE T4
(
    Roll_no NUMBER(10) NOT NULL,
    name VARCHAR2(50),
    IssueDate DATE,
    book VARCHAR2(10),
    status CHAR(1),
    ts TIMESTAMP
);
```

The results section shows the message "Table created." and "0.00 seconds".

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
CREATE TABLE T3
(
    Roll_no NUMBER(10) NOT NULL,
    name VARCHAR2(50),
    IssueDate DATE,
    book VARCHAR2(10),
    status CHAR(1),
    CONSTRAINT pk1_T3 PRIMARY KEY (Roll_no)
);

CREATE TABLE T4
(
    Roll_no NUMBER(10) NOT NULL,
    name VARCHAR2(50),
    IssueDate DATE,
    book VARCHAR2(10),
    status CHAR(1),
    ts TIMESTAMP
);

INSERT ALL
INTO T3 VALUES (101, 'Sita', TO_DATE('22-Dec-2022', 'DD-MON-YYYY'), 'TOC', 'I')
INTO T3 VALUES (102, 'Rita', TO_DATE('20-Sep-2023', 'DD-MON-YYYY'), 'TOC', 'I')
INTO T3 VALUES (103, 'Gita', TO_DATE('13-Nov-2023', 'DD-MON-YYYY'), 'TOC', 'I')
SELECT * FROM dual;
```

The results section shows the message "Table created." and "0.00 seconds".

Application Express 2.1.0.30.39
Copyright © 1996, 2006, Oracle. All rights reserved.

Type here to search 12:40
10-10-2024

WhatsApp SQL Commands

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

```

 Autocommit   

CREATE TABLE T3(
    roll_no VARCHAR(50),
    issuedate DATE,
    bookK VARCHAR(10),
    status CHAR(1),
    ts TIMESTAMP
);

INSERT ALL
INTO T3 VALUES (101, 'Sita', TO_DATE('22-Dec-2022', 'DD-MON-YYYY'), 'TOC', 'I')
INTO T3 VALUES (102, 'Mita', TO_DATE('20-Sep-2023', 'DD-MON-YYYY'), 'TOC', 'I')
INTO T3 VALUES (103, 'Gita', TO_DATE('12-Nov-2023', 'DD-MON-YYYY'), 'TOC', 'I')
SELECT * FROM dual;

SELECT * FROM T3;
TRUNCATE TABLE T1;
TRUNCATE TABLE T2;
CREATE OR REPLACE TRIGGER library_audit
AFTER DELETE OR INSERT OR UPDATE ON T1
FOR EACH ROW
BEGIN
    IF INSERTING THEN
        INSERT INTO T2 (roll_no, name, issuedate, book, status, ts)
        VALUES (new.roll_no, new.name, new.issuedate, new.book, new.status, SYSTIMESTAMP);
    END IF;
END;


```

Results Explain Describe Saved SQL History

5 row(s) inserted.

0.00 seconds

Language errors Application Express 2.1.0.0.39 Copyright © 1999-2008, Oracle. All rights reserved.



WhatsApp SQL Commands

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

```

 Autocommit   

CREATE TABLE T3(
    roll_no VARCHAR(50),
    issuedate DATE,
    bookK VARCHAR(10),
    status CHAR(1),
    ts TIMESTAMP
);

INSERT ALL
INTO T3 VALUES (101, 'Sita', TO_DATE('22-Dec-2022', 'DD-MON-YYYY'), 'TOC', 'I')
INTO T3 VALUES (102, 'Mita', TO_DATE('20-Sep-2023', 'DD-MON-YYYY'), 'TOC', 'I')
INTO T3 VALUES (103, 'Gita', TO_DATE('12-Nov-2023', 'DD-MON-YYYY'), 'TOC', 'I')
SELECT * FROM dual;

SELECT * FROM T3;
TRUNCATE TABLE T1;
TRUNCATE TABLE T2;
CREATE OR REPLACE TRIGGER library_audit
AFTER DELETE OR INSERT OR UPDATE ON T1
FOR EACH ROW
BEGIN
    IF INSERTING THEN
        INSERT INTO T2 (roll_no, name, issuedate, book, status, ts)
        VALUES (new.roll_no, new.name, new.issuedate, new.book, new.status, SYSTIMESTAMP);
    END IF;
END;


```

Results Explain Describe Saved SQL History

ROLL_NO	NAME	ISSUEDATE	BOOK	STATUS
101	Sita	22-OCT-22	TOC	I
102	Mita	20-SEP-23	TOC	I
103	Gita	12-NOV-23	TOC	I

3 rows returned in 0.01 seconds [CSV Export](#)

Language errors Application Express 2.1.0.0.39 Copyright © 1999-2008, Oracle. All rights reserved.



ORAQUEL Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit: Display: 10 Save Run

```
TRUNCATE TABLE T3;
TRUNCATE TABLE T4;

CREATE OR REPLACE TRIGGER library_audit
AFTER DELETE OR INSERT OR UPDATE ON T3
FOR EACH ROW
BEGIN
    IF INSERTING THEN
        INSERT INTO T2 (roll_no, name, issuedate, book, status, ts)
        VALUES (:new.roll_no, :new.name, :new.issuedate, :new.book, :new.status, SYSTIMESTAMP);
    END IF;

    IF UPDATING THEN
        INSERT INTO T2 (roll_no, name, issuedate, book, status, ts)
        VALUES (:old.roll_no, :old.name, :old.issuedate, :old.book, :old.status, SYSTIMESTAMP);
        INSERT INTO T2 (roll_no, name, issuedate, book, status, ts)
    END IF;

```

Results Explain Describe Saved SQL History

Table truncated.

0.00 seconds

Application Express 2.1.0.0.39
Copyright © 1999-2008, Oracle. All rights reserved.



ORAQUEL Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit: Display: 10 Save Run

```
END IF;

IF UPDATING THEN
    INSERT INTO T4 (roll_no, name, issuedate, book, status, ts)
    VALUES (:old.roll_no, :old.name, :old.issuedate, :old.book, :old.status, SYSTIMESTAMP);

    INSERT INTO T4 (roll_no, name, issuedate, book, status, ts)
    VALUES (:new.roll_no, :new.name, :new.issuedate, :new.book, :new.status, SYSTIMESTAMP);
END IF;

IF DELETING THEN
    INSERT INTO T4 (roll_no, name, issuedate, book, status, ts)
    VALUES (:old.roll_no, :old.name, :old.issuedate, :old.book, :old.status, SYSTIMESTAMP);
END IF;

INSERT INTO T1 VALUES (104, 'ABCD', TO_DATE('12-AUG-2024', 'DD-MON-YYYY'), 'AS', 'I');
```

Results Explain Describe Saved SQL History

Trigger created.

0.02 seconds

Application Express 2.1.0.0.39
Copyright © 1999-2008, Oracle. All rights reserved.



The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
END;
/
INSERT INTO T3 VALUES (104, 'ABCD', TO_DATE('12-AUG-2024', 'DD-MON-YYYY'), 'A1', 'I');
SELECT * FROM T3;
SELECT * FROM T4;
UPDATE T3 SET name='Nita' WHERE roll_no=104;
SELECT * FROM T4;
DELETE FROM T3 WHERE roll_no=104;
SELECT * FROM T3;
SELECT * FROM T4;
```

The results section shows:

1 row(s) inserted.

0.00 seconds

Language: en-US Application Express 2.1.0.0.39 Copyright © 1999-2008, Oracle. All rights reserved.

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is identical to the one in the previous screenshot.

The results section shows:

ROLL_NO	NAME	ISSUEDATE	BOOK	STATUS
104	ABCD	12-AUG-24	A1	I

1 rows returned in 0.00 seconds [CSV Export](#)

Language: en-US Application Express 2.1.0.0.39 Copyright © 1999-2008, Oracle. All rights reserved.



WhatsApp SQL Commands

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit: Display: 10 Save Run

```
END;
/
INSERT INTO T3 VALUES (104, 'ABCD', TO_DATE('12-AUG-2024', 'DD-MON-YYYY'), 'A1', 'I');
SELECT * FROM T3;
SELECT * FROM T4;
UPDATE T3 SET name='Nita' WHERE roll_no=104;
SELECT * FROM T4;
DELETE FROM T3 WHERE roll_no=104;
SELECT * FROM T3;
SELECT * FROM T4;
```

Results Explain Describe Saved SQL History

ROLL_NO	NAME	ISSUEDATE	BOOK	STATUS	T5
104	ABCD	12-AUG-24	A1	I	07-OCT-24 12:46:51.020006 PM

1 rows returned in 0.00 seconds CSV Export

Language: en-US Application Express 2.1.0.39 Copyright © 1999-2008, Oracle. All rights reserved.

WhatsApp SQL Commands

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit: Display: 10 Save Run

```
END;
/
INSERT INTO T3 VALUES (104, 'ABCD', TO_DATE('12-AUG-2024', 'DD-MON-YYYY'), 'A1', 'I');
SELECT * FROM T3;
SELECT * FROM T4;
UPDATE T3 SET name='Nita' WHERE roll_no=104;
SELECT * FROM T4;
DELETE FROM T3 WHERE roll_no=104;
SELECT * FROM T3;
SELECT * FROM T4;
```

Results Explain Describe Saved SQL History

1 row(s) updated.

0.00 seconds

Language: en-US Application Express 2.1.0.39 Copyright © 1999-2008, Oracle. All rights reserved.



The screenshot shows the Oracle Database Express Edition SQL Commands window. The SQL code entered is:

```
END;
/
INSERT INTO T3 VALUES (104, 'ABCD', TO_DATE('12-AUG-2024', 'DD-MON-YYYY'), 'A1', '1');
SELECT * FROM T3;
SELECT * FROM T4;
UPDATE T3 SET name='Nita' WHERE roll_no=104;
SELECT * FROM T4;
DELETE FROM T3 WHERE roll_no=104;
SELECT * FROM T3;
SELECT * FROM T4;
```

The results section displays the updated data in a table:

ROLL_NO	NAME	ISSUEDATE	BOOK	STATUS	T\$
104	ABCD	12-AUG-24	A1	I	07-OCT-24 12:46:51 020000 PM
104	ABCD	12-AUG-24	A1	I	07-OCT-24 12:47:47 172000 PM
104	Nita	12-AUG-24	A1	I	07-OCT-24 12:47:47 172000 PM

3 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999-2006, Oracle. All rights reserved.

The screenshot shows the Oracle Database Express Edition SQL Commands window. The SQL code entered is identical to the one in the previous screenshot:

```
END;
/
INSERT INTO T3 VALUES (104, 'ABCD', TO_DATE('12-AUG-2024', 'DD-MON-YYYY'), 'A1', '1');
SELECT * FROM T3;
SELECT * FROM T4;
UPDATE T3 SET name='Nita' WHERE roll_no=104;
SELECT * FROM T4;
DELETE FROM T3 WHERE roll_no=104;
SELECT * FROM T3;
SELECT * FROM T4;
```

The results section shows the deleted row:

1 row(s) deleted.

0.01 seconds

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999-2006, Oracle. All rights reserved.

The screenshot shows the Oracle Database Express Edition SQL Commands window. The SQL code entered is identical to the ones in the previous screenshots:

```
END;
/
INSERT INTO T3 VALUES (104, 'ABCD', TO_DATE('12-AUG-2024', 'DD-MON-YYYY'), 'A1', '1');
SELECT * FROM T3;
SELECT * FROM T4;
UPDATE T3 SET name='Nita' WHERE roll_no=104;
SELECT * FROM T4;
DELETE FROM T3 WHERE roll_no=104;
SELECT * FROM T3;
SELECT * FROM T4;
```

The results section displays the updated data in a table:

ROLL_NO	NAME	ISSUEDATE	BOOK	STATUS	T\$
104	ABCD	12-AUG-24	A1	I	07-OCT-24 12:46:51 020000 PM
104	ABCD	12-AUG-24	A1	I	07-OCT-24 12:47:47 172000 PM
104	Nita	12-AUG-24	A1	I	07-OCT-24 12:47:47 172000 PM

3 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999-2006, Oracle. All rights reserved.

User SYSTEM
Home > SQL > SQL Commands

```
END;
/
INSERT INTO T3 VALUES (104, 'ABCD', TO_DATE('12-AUG-2024', 'DD-MON-YYYY'), 'A1', 'I');
SELECT * FROM T3;
SELECT * FROM T4;
UPDATE T3 SET name='Nita' WHERE roll_no=104;
SELECT * FROM T4;
DELETE FROM T3 WHERE roll_no=104;
SELECT * FROM T3;
SELECT * FROM T4;
```

Results Explain Describe Saved SQL History

no data found

Language: en-US Application Express 2.1.0.0.39 Copyright © 1999-2008, Oracle. All rights reserved.

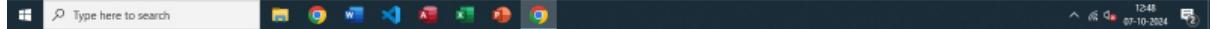
User SYSTEM
Home > SQL > SQL Commands

```
END;
/
INSERT INTO T3 VALUES (104, 'ABCD', TO_DATE('12-AUG-2024', 'DD-MON-YYYY'), 'A1', 'I');
SELECT * FROM T3;
SELECT * FROM T4;
UPDATE T3 SET name='Nita' WHERE roll_no=104;
SELECT * FROM T4;
DELETE FROM T3 WHERE roll_no=104;
SELECT * FROM T3;
SELECT * FROM T4;
```

Results Explain Describe Saved SQL History

ROLL_NO	NAME	ISSUEDATE	BOOK	STATUS	T5
104	ABCD	12-AUG-24	A1	I	07-OCT-24 12:46:51 020000 PM
104	ABCD	12-AUG-24	A1	I	07-OCT-24 12:47:47 172000 PM
104	Nita	12-AUG-24	A1	I	07-OCT-24 12:47:47 172000 PM
104	Nita	12-AUG-24	A1	I	07-OCT-24 12:48:18 652000 PM

4 rows returned in 0.00 seconds CSV Export



Assignment 7

Input:-

```
CREATE TABLE O_ROLL
(
    rollno NUMBER(10) NOT NULL,
    name  VARCHAR2(50),
    gender CHAR(1),
    CONSTRAINT pk1_o PRIMARY KEY (rollno)
);

CREATE TABLE N_ROLL
(
    rollno NUMBER(10) NOT NULL,
    name  VARCHAR2(50),
    gender CHAR(1)
);

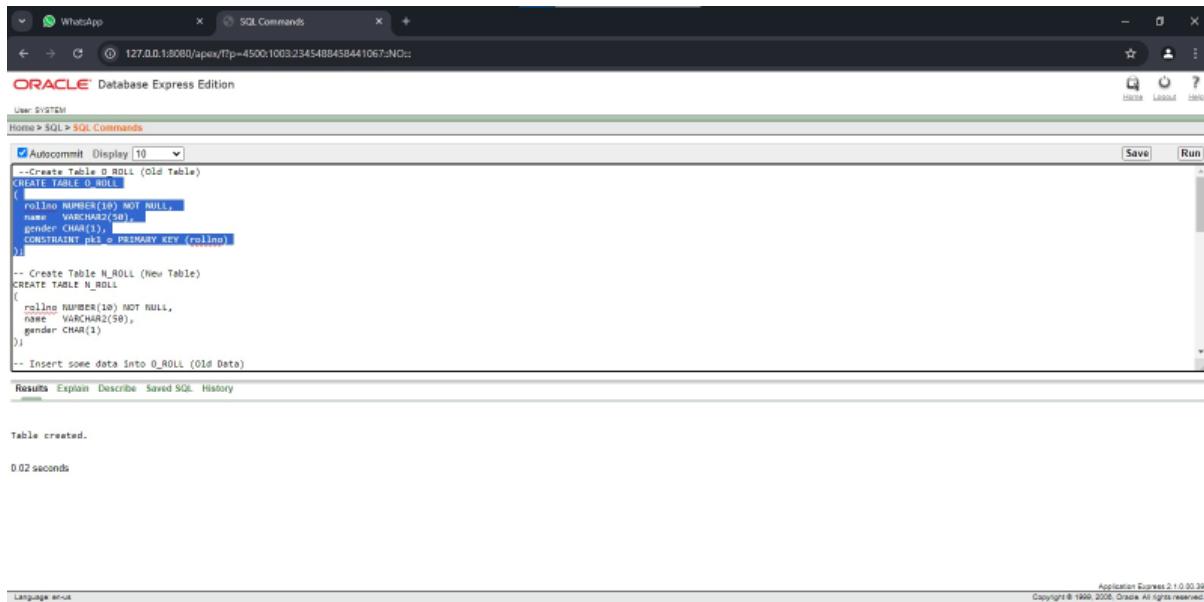
-- Insert some data into O_ROLL (Old Data)
INSERT INTO O_ROLL VALUES (2001, 'xyz', 'M');
INSERT INTO O_ROLL VALUES (2002, 'pqr', 'F');
INSERT INTO O_ROLL VALUES (2005, 'abc', 'F');

-- Insert some data into N_ROLL (New Data)
INSERT INTO N_ROLL VALUES (2001, 'xyz', 'M');

-- PL/SQL Block using an Explicit Cursor to Merge Data from O_ROLL to N_ROLL
DECLARE
    a N_ROLL.rollno%type;
    b N_ROLL.name%type;
    c N_ROLL.gender%type;
    n NUMBER;
CURSOR rollcall IS
    SELECT rollno, name, gender FROM O_ROLL;
```

```
BEGIN  
  OPEN rollcall;  
  LOOP  
    FETCH rollcall INTO a, b, c;  
    EXIT WHEN rollcall%NOTFOUND; -- Exit when all rows have been fetched  
    -- Check if the record already exists in N_ROLL  
    SELECT COUNT(*) INTO n FROM N_ROLL WHERE rollno = a;  
    IF n > 0 THEN  
      dbms_output.put_line('Data already exists for Roll No: ' || a);  
    ELSE  
      INSERT INTO N_ROLL VALUES (a, b, c);  
      dbms_output.put_line('Inserted Roll No: ' || a);  
    END IF;  
  END LOOP;  
  CLOSE rollcall;  
END;  
/
```

Output:-



The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
--Create Table O_ROLL (Old Table)
CREATE TABLE O_ROLL
(
    rollno NUMBER(10) NOT NULL,
    name VARCHAR2(50),
    gender CHAR(1),
    CONSTRAINT pk1_o PRIMARY KEY (rollno)
);

-- Create Table N_ROLL (New Table)
CREATE TABLE N_ROLL
(
    rollno NUMBER(10) NOT NULL,
    name VARCHAR2(50),
    gender CHAR(1)
);

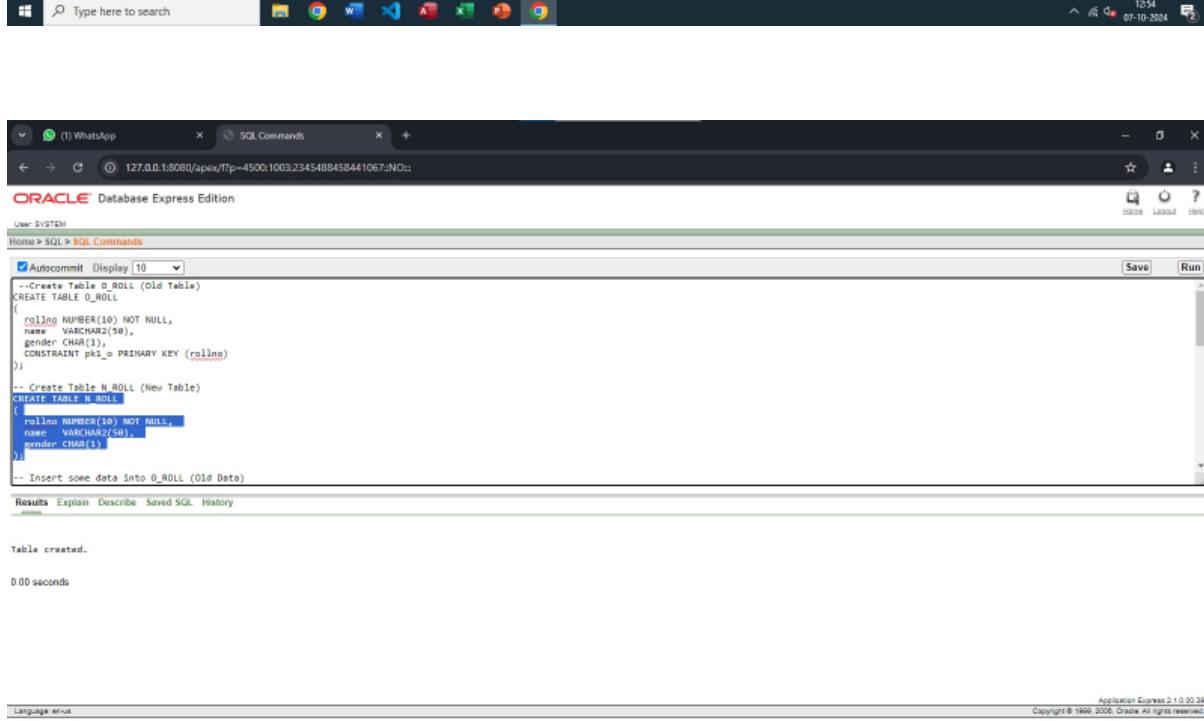
-- Insert some data into O_ROLL (Old Data)

```

The results pane shows the output of the SQL commands:

```
Table created.
0.02 seconds
```

At the bottom right, it says "Application Express 2.1.0.33.39 Copyright © 1999-2006, Oracle. All rights reserved."



The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is identical to the one in the previous screenshot:

```
--Create Table O_ROLL (Old Table)
CREATE TABLE O_ROLL
(
    rollno NUMBER(10) NOT NULL,
    name VARCHAR2(50),
    gender CHAR(1),
    CONSTRAINT pk1_o PRIMARY KEY (rollno)
);

-- Create Table N_ROLL (New Table)
CREATE TABLE N_ROLL
(
    rollno NUMBER(10) NOT NULL,
    name VARCHAR2(50),
    gender CHAR(1)
);

-- Insert some data into O_ROLL (Old Data)

```

The results pane shows the output of the SQL commands:

```
Table created.
0.00 seconds
```

At the bottom right, it says "Application Express 2.1.0.33.39 Copyright © 1999-2006, Oracle. All rights reserved."



The screenshot shows the Oracle Database Express Edition SQL Commands interface. The code entered is:

```
-- Insert some data into O_ROLL (Old Data)
INSERT INTO O_ROLL VALUES (2001, 'xyz', 'M');
INSERT INTO O_ROLL VALUES (2002, 'abc', 'F');
INSERT INTO O_ROLL VALUES (2003, 'abc', 'F');

-- Insert some data into N_ROLL (New Data)
INSERT INTO N_ROLL VALUES (2001, 'xyz', 'M');

-- PL/SQL Block using an Explicit Cursor to Merge Data from O_ROLL to N_ROLL
DECLARE
  a N_ROLL%ColumnTypes;
  b N_ROLL.name%Type;
  c N_ROLL.gender%Type;
  n NUMBER;
```

The results section shows:

1 row(s) inserted.
0.00 seconds

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999-2008, Oracle. All rights reserved.

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The code entered is identical to the one in the first screenshot:

```
-- Insert some data into O_ROLL (Old Data)
INSERT INTO O_ROLL VALUES (2001, 'xyz', 'M');
INSERT INTO O_ROLL VALUES (2002, 'abc', 'F');
INSERT INTO O_ROLL VALUES (2003, 'abc', 'F');

-- Insert some data into N_ROLL (New Data)
INSERT INTO N_ROLL VALUES (2001, 'xyz', 'M');

-- PL/SQL Block using an Explicit Cursor to Merge Data from O_ROLL to N_ROLL
DECLARE
  a N_ROLL%ColumnTypes;
  b N_ROLL.name%Type;
  c N_ROLL.gender%Type;
  n NUMBER;
```

The results section shows:

1 row(s) inserted.
0.00 seconds

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999-2008, Oracle. All rights reserved.



(1) WhatsApp

SQL Commands

User: SYSTEM

Home > SQL > SQL Commands

Autocommit: Display: 10 Save Run

```
LOOP
  FETCH rollcall INTO a, b, c;
  EXIT WHEN rollcall%NOTFOUND; -- Exit when all rows have been fetched.

  -- Check if the record already exists in N_ROLL
  SELECT COUNT(*) INTO n FROM N_ROLL WHERE rollno = a;

  IF n > 0 THEN
    dnm_output.put_line('Data already exists for Roll No: ' || a);
  ELSE
    INSERT INTO N_ROLL VALUES (a, b, c);
    dnm_output.put_line('Inserted Roll No: ' || a);
  END IF;
END LOOP;
CLOSE rollcall;
END;
```

Results Explain Describe Saved SQL History

Data already exists for Roll No: 2001
Inserted Roll No: 2002
Inserted Roll No: 2005

1 row(s) inserted.

0.00 seconds

Language: en-US Application Express 2.1.0.0.39 Copyright © 1999-2008, Oracle. All rights reserved.



Assignment 6

Input:-

```
CREATE TABLE X
(
    Roll_no NUMBER(10) NOT NULL,
    name  VARCHAR2(50),
    Marks  NUMBER(10),
    CONSTRAINT pk1_x PRIMARY KEY (Roll_no)
);

CREATE TABLE Y
(
    Roll_no NUMBER(10) NOT NULL,
    name  VARCHAR2(50),
    marks  NUMBER(10),
    result VARCHAR2(50)
);

INSERT INTO X VALUES (2001, 'xyz', 999);
INSERT INTO X VALUES (2002, 'pqr', 500);

CREATE OR REPLACE PROCEDURE Grade(rnum IN NUMBER) AS
    mrks  NUMBER;
    s_name VARCHAR2(50);
    GRADE VARCHAR2(50);

BEGIN
    SELECT name, marks INTO s_name, mrks FROM X WHERE Roll_no = rnum;
    IF mrks <= 1500 AND mrks >= 990 THEN
        GRADE := 'DISTINCTION';
    ELSIF mrks <= 989 AND mrks >= 900 THEN
        GRADE := 'FIRST CLASS';
    ELSIF mrks <= 899 AND mrks >= 825 THEN
```

```
GRADE := 'HIGHER SECOND CLASS';
ELSE
    GRADE := 'PASS CLASS';
END IF;
INSERT INTO Y VALUES (rnum, s_name, mrks, GRADE);
dbms_output.put_line('ROLL_No  NAME  MARKS  GRADE');
dbms_output.put_line(" || rnum || chr(9) || s_name || chr(9) || mrks || chr(9) || GRADE");
END;
/
DECLARE
rollnumber NUMBER;
BEGIN
rollnumber := 2001;
dbms_output.put_line('ENTER THE ROLL NUMBER OF THE STUDENT: ' || rollnumber);
Grade(rollnumber);
END;
/
SELECT * FROM Y;
SELECT * FROM X;
```

Output:-

CREATE TABLE X
(Roll_no NUMBER(10) NOT NULL,
name VARCHAR2(50),
Marks NUMBER(10),
CONSTRAINT pk1 PRIMARY KEY (Roll_no)
);

CREATE TABLE Y
(Roll_no NUMBER(10) NOT NULL,
name VARCHAR2(50),
marks NUMBER(10),
result VARCHAR2(50)
);

INSERT INTO X VALUES (2001, 'xyz', 999);
INSERT INTO X VALUES (2002, 'pqr', 500);

Table created.
0.00 seconds

Language: en-US Application Express 2.1.0.33.39 Copyright © 1999, 2006, Oracle. All rights reserved.

CREATE OR REPLACE PROCEDURE Grade(rnum IN NUMBER) AS
marks NUMBER;
s_name VARCHAR2(50);
GRADE VARCHAR2(50);
BEGIN
SELECT name, marks INTO s_name, marks FROM X WHERE Roll_no = rnum;
IF marks <= 1500 AND marks >= 900 THEN
GRADE := 'DISTINCTION';
ELSIF marks <= 899 AND marks >= 900 THEN
GRADE := 'FIRST CLASS';
ELSIF marks <= 898 AND marks >= 625 THEN
GRADE := 'SECOND CLASS';
END IF;

1 row(s) inserted.
0.00 seconds

Language: en-US Application Express 2.1.0.33.39 Copyright © 1999, 2006, Oracle. All rights reserved.



```
ORA- Database Express Edition
User: SYSTEM
Home > SQL > SQL Commands
Autocommit: Display: 10 | Save | Run
Result: VARCHAR2(50)
;
INSERT INTO X VALUES (2001, 'xyz', 999);
INSERT INTO X VALUES (2002, 'yqr', 500);
CREATE OR REPLACE PROCEDURE Grade(rnum IN NUMBER) AS
  mks NUMBER;
  s_name VARCHAR2(50);
  GRADE VARCHAR2(50);
BEGIN
  SELECT name, marks INTO s_name, mks FROM X WHERE Roll_no = rnum;
  IF mks <= 1500 AND mks >= 900 THEN
    GRADE := 'DISTINCTION';
  ELSIF mks <= 900 AND mks >= 800 THEN
    GRADE := 'FIRST CLASS';
  ELSIF mks <= 800 AND mks >= 625 THEN
    GRADE := 'SECOND CLASS';
  ELSE
    GRADE := 'FAIL';
  END IF;
  dbms_output.put_line('ROLL_NO NAME MARKS GRADE');
  dbms_output.put_line(rnum || ' ' || s_name || ' ' || mks || ' ' || GRADE);
END;
/
Results Explain Describe Saved SQL History
```

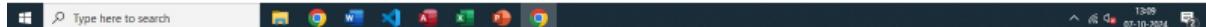
1 row(s) inserted.
0.00 seconds

Application Express 2.1.0.0.39
Copyright © 1999-2008, Oracle. All rights reserved.

```
ORA- Database Express Edition
User: SYSTEM
Home > SQL > SQL Commands
Autocommit: Display: 10 | Save | Run
Result: VARCHAR2(50)
;
INSERT INTO X VALUES (rnum, s_name, mks, GRADE);
dbms_output.put_line("ROLL_NO NAME MARKS GRADE");
dbms_output.put_line(rnum || ' ' || s_name || ' ' || mks || ' ' || GRADE);
END;
/
DECLARE
  rollNumber NUMBER;
BEGIN
  rollNumber := 2001;
  dbms_output.put_line("ENTER THE ROLL NUMBER OF THE STUDENT: " || rollNumber);
  Grade(rollNumber);
END;
/
SELECT * FROM Y;
SELECT * FROM Z;
```

Procedure created.
0.02 seconds

Application Express 2.1.0.0.39
Copyright © 1999-2008, Oracle. All rights reserved.



ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
INSERT INTO Y VALUES (rnum, s_name, marks, GRADE);
dbms_output.put_line('ROLL_No NAME MARKS GRADE');
dbms_output.put_line(' ' || rnum || chr(9) || s_name || chr(9) || marks || chr(9) || GRADE);
END;
/
DECLARE
  rollnumber NUMBER;
BEGIN
  rollnumber := 2001;
  dbms_output.put_line("ENTER THE ROLL NUMBER OF THE STUDENT: " || rollnumber);
  grade(rollnumber);
END;
/
SELECT * FROM Y;
SELECT * FROM X;
```

Results Explain Describe Saved SQL History

ENTER THE ROLL NUMBER OF THE STUDENT: 2001

ROLL_No	NAME	MARKS	GRADE
2001	xyz	999	DISTINCTION

Statement processed.

0.00 seconds

Language errors Application Express 2.1.0.0.39 Copyright © 1999-2008, Oracle. All rights reserved.



ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
INSERT INTO Y VALUES (rnum, s_name, marks, GRADE);
dbms_output.put_line('ROLL_No NAME MARKS GRADE');
dbms_output.put_line(' ' || rnum || chr(9) || s_name || chr(9) || marks || chr(9) || GRADE);
END;
/
DECLARE
  rollnumber NUMBER;
BEGIN
  rollnumber := 2001;
  dbms_output.put_line("ENTER THE ROLL NUMBER OF THE STUDENT: " || rollnumber);
  grade(rollnumber);
END;
/
SELECT * FROM Y;
SELECT * FROM X;
```

Results Explain Describe Saved SQL History

ROLL_No	NAME	MARKS	RESULT
2001	xyz	999	DISTINCTION

1 rows returned in 0.00 seconds CSV Export

Language errors Application Express 2.1.0.0.39 Copyright © 1999-2008, Oracle. All rights reserved.



WhatsApp SQL Commands

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

Autocommit: Display: 10 Save Run

```
INSERT INTO Y VALUES (rnum, s_name, marks, GRADE);
dbms_output.put_line('ROLL_NO NAME MARKS GRADE');
dbms_output.put_line('' || rnum || chr(9) || s_name || chr(9) || marks || chr(9) || GRADE);
END;
/
DECLARE
rollnumber NUMBER;
BEGIN
rollnumber := 2001;
dbms_output.put_line("ENTER THE ROLL NUMBER OF THE STUDENT: " || rollnumber);
Grade(rollnumber);
END;
/
SELECT * FROM Y;
SELECT * FROM X;
```

Results Explain Describe Saved SQL History

ROLL_NO	NAME	MARKS
2001	xyz	999
2002	pqr	999

2 rows returned in 0.00 seconds CSV Export

Language: sqlplus Application Express 2.1.0.0.39 Copyright © 1999-2008, Oracle. All rights reserved.



Assignment 5

Input:-

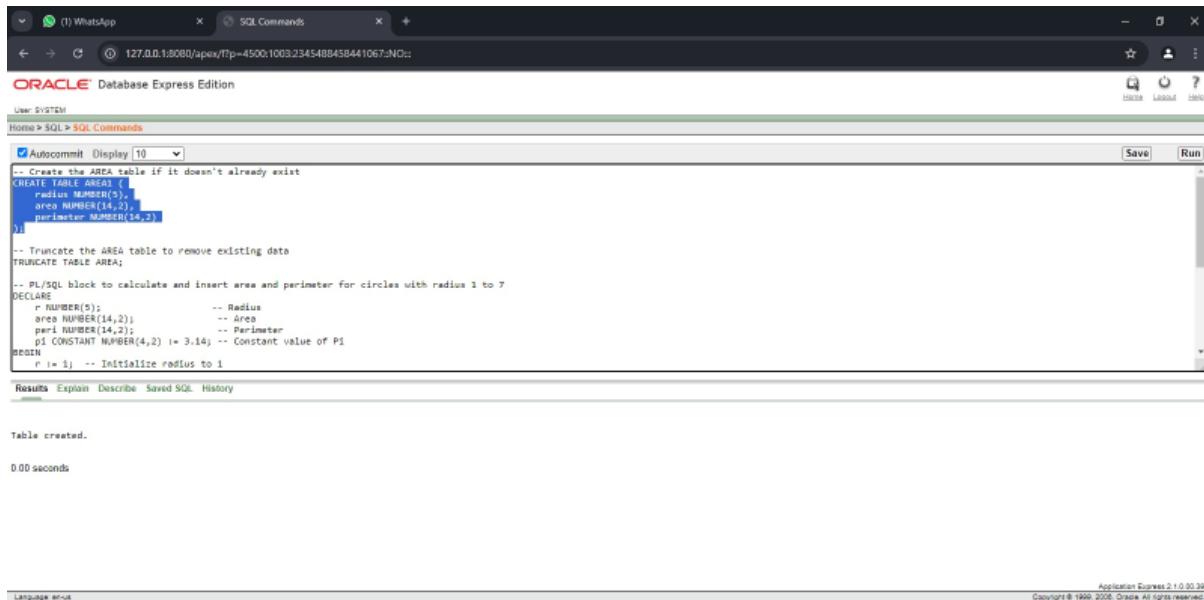
```
-- Create the AREA table if it doesn't already exist
CREATE TABLE AREA (
    radius NUMBER(5),
    area NUMBER(14,2),
    perimeter NUMBER(14,2)
);

-- Truncate the AREA table to remove existing data
TRUNCATE TABLE AREA;

-- PL/SQL block to calculate and insert area and perimeter for circles with radius 1 to 7
DECLARE
    r NUMBER(5);          -- Radius
    area NUMBER(14,2);     -- Area
    peri NUMBER(14,2);      -- Perimeter
    pi CONSTANT NUMBER(4,2) := 3.14; -- Constant value of Pi
BEGIN
    r := 1; -- Initialize radius to 1
    WHILE r <= 7 LOOP
        -- Calculate area and perimeter
        area := pi * POWER(r, 2);
        peri := 2 * pi * r;
        -- Insert into AREA table
        INSERT INTO AREA (radius, area, perimeter) VALUES (r, area, peri);
        -- Increment the radius for the next iteration
        r := r + 1;
    END LOOP;
END;
```

SELECT * FROM AREA;

Outputs:-



The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following PL/SQL script is run:

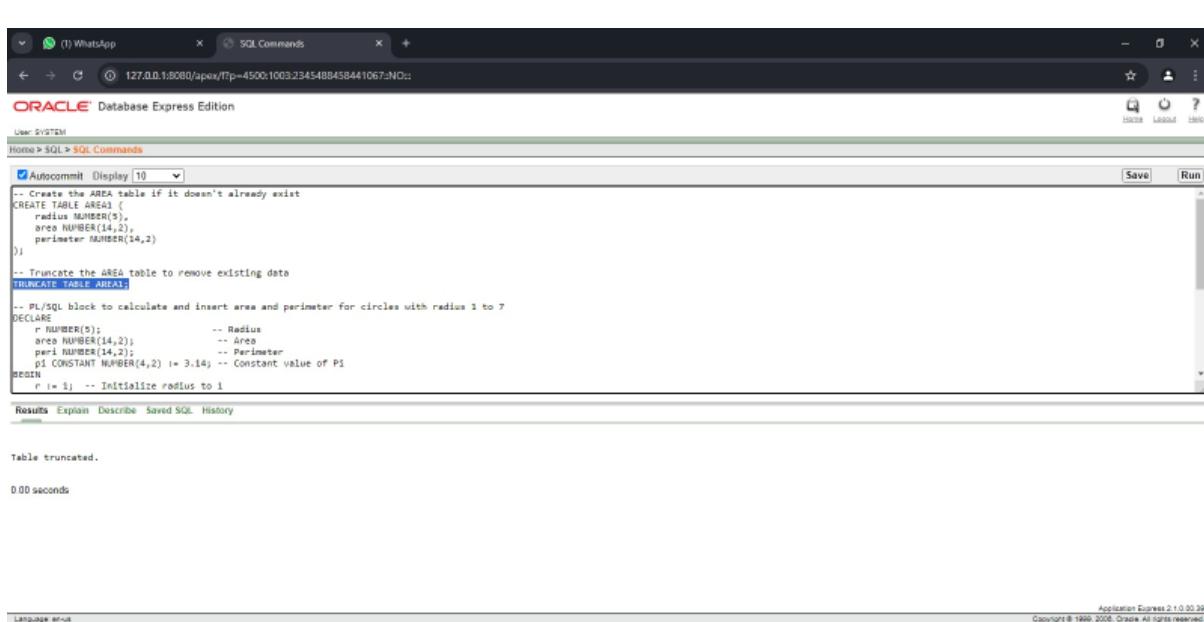
```
-- Create the AREA table if it doesn't already exist
CREATE TABLE AREA1 (
    radius NUMBER(5),
    area NUMBER(14,2),
    perimeter NUMBER(14,2)
);

-- Truncate the AREA table to remove existing data
TRUNCATE TABLE AREA1;

-- PL/SQL block to calculate and insert area and perimeter for circles with radius 1 to 7
DECLARE
    r NUMBER(5);          -- Radius
    area NUMBER(14,2);    -- Area
    peri NUMBER(14,2);   -- Perimeter
    pi CONSTANT NUMBER(4,2) := 3.14; -- Constant value of PI
BEGIN
    r := 1; -- Initialize radius to 1
    FOR i IN 1..7
        LOOP
            r := i;
            area := pi * r * r;
            peri := 2 * pi * r;
            INSERT INTO AREA1 (radius, area, perimeter) VALUES (r, area, peri);
        END LOOP;
    END;
END;
```

The results show the table was created and no time was taken.

Language:enus Application Express 2.1.0.33.39 Copyright © 1999-2008, Oracle. All rights reserved.



The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following PL/SQL script is run:

```
-- Create the AREA table if it doesn't already exist
CREATE TABLE AREA1 (
    radius NUMBER(5),
    area NUMBER(14,2),
    perimeter NUMBER(14,2)
);

-- Truncate the AREA table to remove existing data
TRUNCATE TABLE AREA1;

-- PL/SQL block to calculate and insert area and perimeter for circles with radius 1 to 7
DECLARE
    r NUMBER(5);          -- Radius
    area NUMBER(14,2);    -- Area
    peri NUMBER(14,2);   -- Perimeter
    pi CONSTANT NUMBER(4,2) := 3.14; -- Constant value of PI
BEGIN
    r := 1; -- Initialize radius to 1
    FOR i IN 1..7
        LOOP
            r := i;
            area := pi * r * r;
            peri := 2 * pi * r;
            INSERT INTO AREA1 (radius, area, perimeter) VALUES (r, area, peri);
        END LOOP;
    END;
END;
```

The results show the table was truncated and no time was taken.

Language:enus Application Express 2.1.0.33.39 Copyright © 1999-2008, Oracle. All rights reserved.



(1) WhatsApp

SQL Commands

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit: Display: 10 Save Run

```
BEGIN
    r := 1; -- Initialize radius to 1
    WHILE r <= 7 LOOP
        -- Calculate area and perimeter
        area := pi * POWER(r, 2);
        peri := 2 * pi * r;
        
        -- Insert into AREA table
        INSERT INTO AREA (radius, area, perimeter) VALUES (r, area, peri);
        
        -- Increment the radius for the next iteration
        r := r + 1;
    END LOOP;
/
-- Verify results
SELECT * FROM AREA;
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.00 seconds

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999-2008, Oracle. All rights reserved.



(1) WhatsApp

SQL Commands

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit: Display: 10 Save Run

```
BEGIN
    r := 1; -- Initialize radius to 1
    WHILE r <= 7 LOOP
        -- Calculate area and perimeter
        area := pi * POWER(r, 2);
        peri := 2 * pi * r;
        
        -- Insert into AREA table
        INSERT INTO AREA (radius, area, perimeter) VALUES (r, area, peri);
        
        -- Increment the radius for the next iteration
        r := r + 1;
    END LOOP;
/
-- Verify results
SELECT * FROM AREA;
```

Results Explain Describe Saved SQL History

RADIUS	AREA	PERIMETER
1	3.14	6.28
2	12.56	12.56
3	28.26	18.84
4	50.24	25.12
5	78.5	31.4
6	113.04	37.68
7	153.86	43.96

7 rows returned in 0.00 seconds CSV Export

Application Express 2.1.0.0.39 Copyright © 1999-2008, Oracle. All rights reserved.



Assignment 4

Input:-

Assignment 3

Input:-

```
CREATE TABLE College
(
    college_id NUMERIC(3) PRIMARY KEY,
    college_name VARCHAR(15)
);

CREATE TABLE cdepartment
(
    Department_ID NUMERIC(2) PRIMARY KEY,
    college_id NUMERIC(3),
    name VARCHAR(20),
    FOREIGN KEY (college_id) REFERENCES College(college_id)
);

INSERT INTO College VALUES(101,'xyz');

INSERT INTO College VALUES(102,'abc');

INSERT INTO College VALUES(103,'mno');

INSERT INTO cdepartment VALUES(1,101,'xyzp');

INSERT INTO cdepartment VALUES(2,102,'abc2');

-- Corrected college_id for department 'mno3' (from 104 to 103)

INSERT INTO cdepartment VALUES(3,103,'mno3');

-- Step : Verify Data in Both Tables

SELECT * FROM College;

SELECT * FROM cdepartment;

-- 1. INNER JOIN

SELECT College.college_id, College.college_name, cdepartment.department_id, cdepartment.name
FROM College
INNER JOIN cdepartment
ON College.college_id = cdepartment.college_id;
```

```
-- 2. NATURAL JOIN
SELECT *
FROM College
NATURAL JOIN cdepartment;

-- 3. LEFT JOIN
SELECT College.college_id, College.college_name, cdepartment.name
FROM College
LEFT JOIN cdepartment
ON College.college_id = cdepartment.college_id;

-- 4. RIGHT JOIN
SELECT College.college_id, College.college_name, cdepartment.name
FROM College
RIGHT JOIN cdepartment
ON College.college_id = cdepartment.college_id;

-- 5. FULL JOIN
SELECT College.college_name, cdepartment.name
FROM College
FULL JOIN cdepartment
ON College.college_id = cdepartment.college_id
ORDER BY College.college_id;

-- 6. CROSS JOIN
SELECT College.college_name, cdepartment.name
FROM College, cdepartment
WHERE College.college_id = cdepartment.college_id;

-- 7. Sub-query
SELECT college_name
FROM College
WHERE college_id = (SELECT college_id FROM cdepartment WHERE department_id = (SELECT MAX(department_id) FROM cdepartment));
```

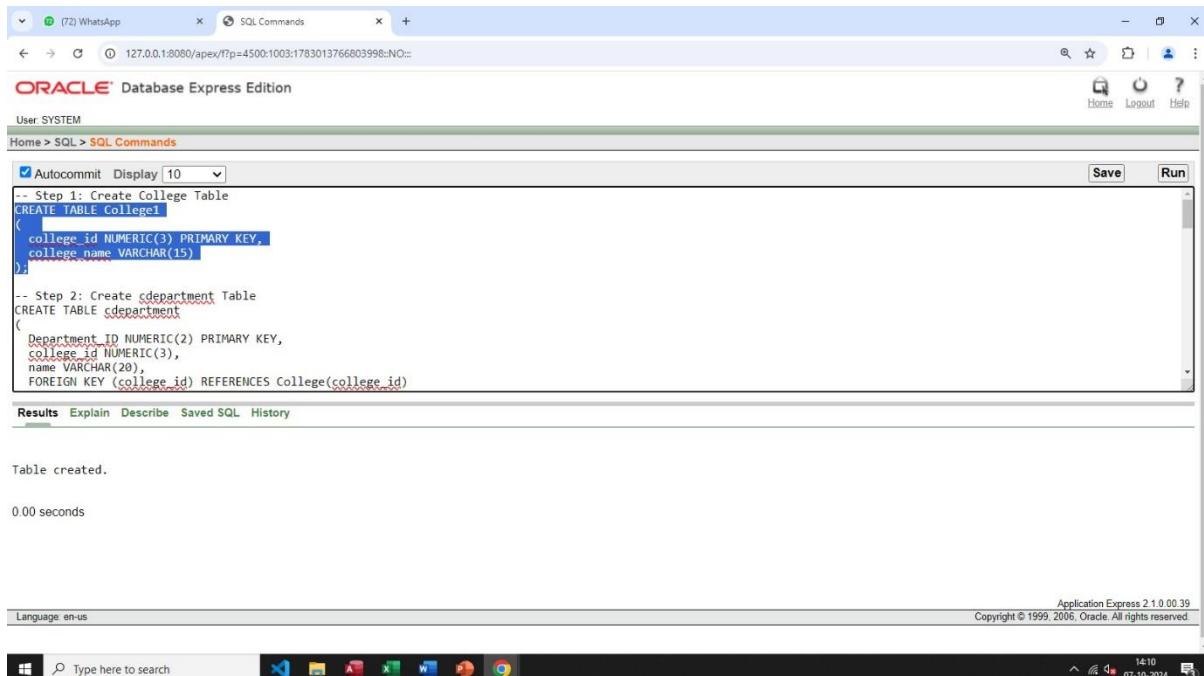
-- 8. Sub-query

```
SELECT name  
FROM cdepartment  
WHERE college_id = (SELECT MIN(college_id) FROM College);
```

-- 9. View

```
CREATE VIEW College_Department_View AS  
SELECT College.college_name, cdepartment.name  
FROM College  
INNER JOIN cdepartment ON College.college_id = cdepartment.college_id;  
-- 10. Query the view  
SELECT * FROM College_Department_View;
```

Output:-

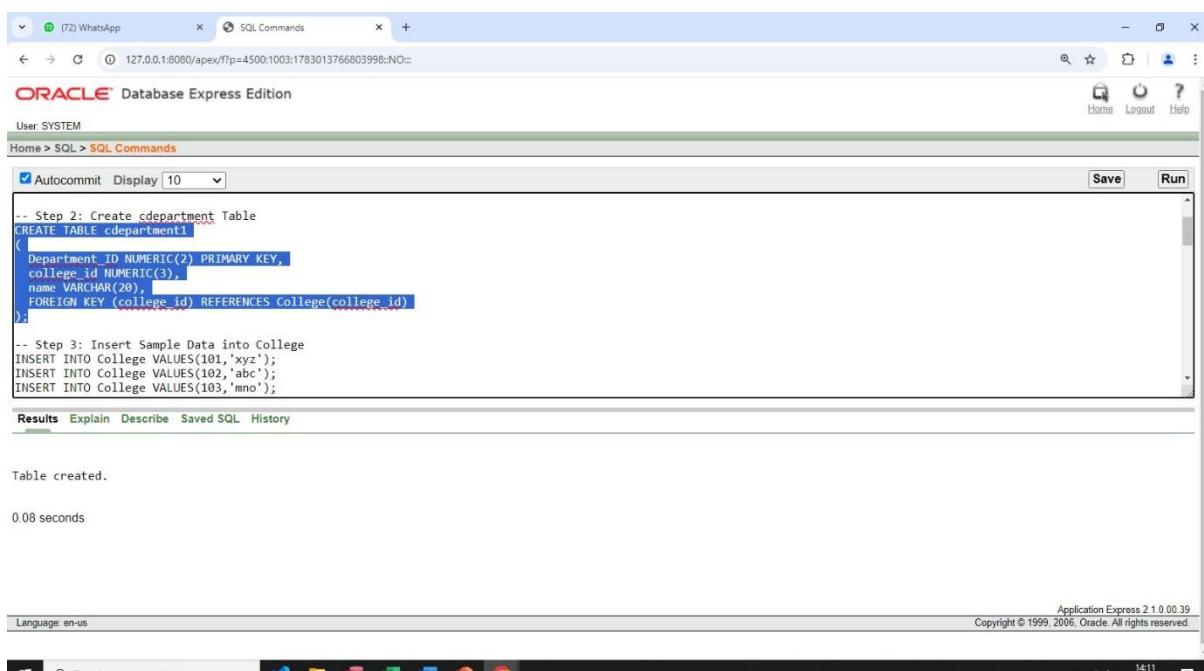


The screenshot shows the Oracle Database Express Edition SQL Commands interface. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:1783013766803998::NO:::. The page title is ORACLE® Database Express Edition. The user is SYSTEM. The SQL editor contains the following code:

```
-- Step 1: Create College Table
CREATE TABLE College
(
    college_id NUMERIC(3) PRIMARY KEY,
    college_name VARCHAR(15)
);

-- Step 2: Create cdepartment Table
CREATE TABLE cdepartment
(
    Department_ID NUMERIC(2) PRIMARY KEY,
    college_id NUMERIC(3),
    name VARCHAR(20),
    FOREIGN KEY (college_id) REFERENCES College(college_id)
);
```

The results show "Table created." and "0.00 seconds". The status bar indicates Application Express 2.1.0.0.39 and Copyright © 1999, 2006, Oracle. All rights reserved.



The screenshot shows the Oracle Database Express Edition SQL Commands interface. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:1783013766803998::NO:::. The page title is ORACLE® Database Express Edition. The user is SYSTEM. The SQL editor contains the following code:

```
-- Step 2: Create cdepartment Table
CREATE TABLE cdepartment
(
    Department_ID NUMERIC(2) PRIMARY KEY,
    college_id NUMERIC(3),
    name VARCHAR(20),
    FOREIGN KEY (college_id) REFERENCES College(college_id)
);

-- Step 3: Insert Sample Data into College
INSERT INTO College VALUES(101,'xyz');
INSERT INTO College VALUES(102,'abc');
INSERT INTO College VALUES(103,'mno');
```

The results show "Table created." and "0.08 seconds". The status bar indicates Application Express 2.1.0.0.39 and Copyright © 1999, 2006, Oracle. All rights reserved.

(72) WhatsApp SQL Commands +

127.0.0.1:8080/apex/f?p=4500:1003:1783013766803998::NO::

ORACLE® Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
College_id NUMERIC(3),
name VARCHAR(20),
FOREIGN KEY (college_id) REFERENCES College(college_id)
);

-- Step 3: Insert Sample Data into College
INSERT INTO College1 VALUES(101,'xyz');
INSERT INTO College1 VALUES(102,'abc');
INSERT INTO College1 VALUES(103,'mno');

-- Step 4: Insert Sample Data into cdepartment
INSERT INTO cdepartment VALUES(1,101,'xyzp');
INSERT INTO cdepartment VALUES(2,102,'abc2');
-- Corrected college_id for department 'mno3' (from 104 to 103)
INSERT INTO cdepartment VALUES(3,103,'mno3');
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.00 seconds

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

(72) WhatsApp SQL Commands +

127.0.0.1:8080/apex/f?p=4500:1003:1783013766803998::NO::

ORACLE® Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
Step 3: Insert Sample Data into College
INSERT INTO College VALUES(101,'xyz');
INSERT INTO College VALUES(102,'abc');
INSERT INTO College VALUES(103,'mno');

-- Step 4: Insert Sample Data into cdepartment
INSERT INTO cdepartment VALUES(1,101,'xyzp');
INSERT INTO cdepartment VALUES(2,102,'abc2');
-- Corrected college_id for department 'mno3' (from 104 to 103)
INSERT INTO cdepartment VALUES(3,103,'mno3');

-- Step 5: Verify Data in Both Tables
SELECT * FROM College;
SELECT * FROM cdepartment;
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.00 seconds

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.



ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```

Step 3: Insert Sample Data into College
INSERT INTO College1 VALUES(101,'xyz');
INSERT INTO College1 VALUES(102,'abc');
INSERT INTO College1 VALUES(103,'mno');

-- Step 4: Insert Sample Data into cdepartment
INSERT INTO cdepartment1 VALUES(1,101,'xyzp');
INSERT INTO cdepartment1 VALUES(2,102,'abc2');
-- Corrected college_id for department 'mno3' (from 104 to 103)
INSERT INTO cdepartment1 VALUES(3,103,'mno3');

-- Step 5: Verify Data in Both Tables
SELECT * FROM College;
SELECT * FROM cdepartment;

```

1 row(s) inserted.

0.00 seconds

Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.



ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```

INSERT INTO cdepartment1 VALUES(1,101,'xyzp');
INSERT INTO cdepartment1 VALUES(2,102,'abc2');
-- Corrected college_id for department 'mno3' (from 104 to 103)
INSERT INTO cdepartment1 VALUES(3,103,'mno3');

-- Step 5: Verify Data in Both Tables
SELECT * FROM College;
SELECT * FROM cdepartment;

-- 1. INNER JOIN: Retrieves rows with matching college_id in both tables
SELECT College.college_id, College.college_name, cdepartment.department_id, cdepartment.name
FROM College
INNER JOIN cdepartment
ON College.college_id = cdepartment.college_id;

```

COLLEGE_ID	COLLEGE_NAME
101	xyz
102	abc
103	mno

3 rows returned in 0.00 seconds [CSV Export](#)



(7Z WhatsApp) SQL Commands

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO cdepartment1 VALUES(1,101,'xyzp');
INSERT INTO cdepartment1 VALUES(2,102,'abc2');
-- Corrected college_id for department 'mno3' (from 104 to 103)
INSERT INTO cdepartment1 VALUES(3,103,'mno3');

-- Step 5: Verify Data in Both Tables
SELECT * FROM College1;

-- 1. INNER JOIN: Retrieves rows with matching college_id in both tables
SELECT College.college_id, College.college_name, cdepartment.department_id, cdepartment.name
FROM College
INNER JOIN cdepartment
ON College.college_id = cdepartment.college_id;
```

Results Explain Describe Saved SQL History

DEPARTMENT_ID	COLLEGE_ID	NAME
1	101	xyzp
2	102	abc2
3	103	mno3

3 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999-2010, Oracle. All rights reserved.

Type here to search 14:14 07-10-2024

(7Z WhatsApp) SQL Commands

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
-- Step 5: Verify Data in Both Tables
SELECT * FROM College1;
SELECT * FROM cdepartment1;

-- 1. INNER JOIN: Retrieves rows with matching college_id in both tables
SELECT College1.college_id, College1.college_name, cdepartment1.department_id, cdepartment1.name
FROM College1
INNER JOIN cdepartment1
ON College1.college_id = cdepartment1.college_id;

-- 2. NATURAL JOIN: Automatically joins on columns with the same name
SELECT *
FROM College
NATURAL JOIN cdepartment;
```

Results Explain Describe Saved SQL History

COLLEGE_ID	COLLEGE_NAME	DEPARTMENT_ID	NAME
101	xyz	1	xyzp
102	abc	2	abc2
103	mno	3	mno3

3 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999-2010, Oracle. All rights reserved.

Type here to search 14:16 07-10-2024

(7Z WhatsApp) SQL Commands

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
SELECT college1.college_id, college1.college_name, cdepartment1.department_id, cdepartment1.name
FROM College1
INNER JOIN cdepartment1
ON College1.college_id = cdepartment1.college_id;

-- 2. NATURAL JOIN: Automatically joins on columns with the same name
SELECT *
FROM College1
NATURAL JOIN cdepartment1;

-- 3. LEFT JOIN: Retrieves all rows from College, with NULLs for non-matching departments
SELECT College1.college_id, College1.college_name, cdepartment1.name
FROM College1
LEFT JOIN cdepartment1
ON College1.college_id = cdepartment1.college_id;
```

Results Explain Describe Saved SQL History

COLLEGE_ID	COLLEGE_NAME	DEPARTMENT_ID	NAME
101	xyz	1	xyzp
102	abc	2	abc2
103	mno	3	mno3

3 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1 0 0 39 Copyright © 1999-2006, Oracle. All rights reserved.

Type here to search 14:16 07-10-2024

(7Z WhatsApp) SQL Commands

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
-- 2. NATURAL JOIN: Automatically joins on columns with the same name
SELECT *
FROM College1
NATURAL JOIN cdepartment1;

-- 3. LEFT JOIN: Retrieves all rows from College, with NULLs for non-matching departments
SELECT College1.college_id, College1.college_name, cdepartment1.name
FROM College1
LEFT JOIN cdepartment1
ON College1.college_id = cdepartment1.college_id;

-- 4. RIGHT JOIN: Retrieves all rows from cdepartment1, with NULLs for non-matching colleges
SELECT College1.college_id, College1.college_name, cdepartment1.name
FROM College1
RIGHT JOIN cdepartment1
```

Results Explain Describe Saved SQL History

COLLEGE_ID	COLLEGE_NAME	NAME
101	xyz	xyzp
102	abc	abc2
103	mno	mno3

3 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1 0 0 39 Copyright © 1999-2006, Oracle. All rights reserved.

Type here to search 14:17 07-10-2024

(72) WhatsApp SQL Commands 127.0.0.1:8080/apex/f?p=4500:1003:1783013766803998::NO::

ORACLE® Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
-- 4. RIGHT JOIN: Retrieves all rows from cdepartment, with NULLs for non-matching colleges
SELECT College1.college_id, College1.college_name, cdepartment1.name
FROM College1
RIGHT JOIN cdepartment1
ON College1.college_id = cdepartment1.college_id;

-- 5. FULL JOIN: Retrieves all rows from both tables, with NULLs where there is no match
SELECT College1.college_name, cdepartment1.name
FROM College1
FULL JOIN cdepartment1
ON college.college_id = cdepartment.college_id
ORDER BY College1.college_id;

-- 6. CROSS JOIN: Cartesian product (every row in College combined with every row in cdepartment)
```

COLLEGE_ID	COLLEGE_NAME	NAME
101	xyz	xyzp
102	abc	abc2
103	mno	mno3

3 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1 0.0.39 Copyright © 1999-2006, Oracle. All rights reserved.

Type here to search 14:18 07-10-2024

(72) WhatsApp SQL Commands 127.0.0.1:8080/apex/f?p=4500:1003:1783013766803998::NO::

ORACLE® Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
-- 4. RIGHT JOIN: Retrieves all rows from cdepartment, with NULLs for non-matching colleges
SELECT College1.college_id, College1.college_name, cdepartment1.name
FROM College1
RIGHT JOIN cdepartment1
ON College1.college_id = cdepartment1.college_id;

-- 5. FULL JOIN: Retrieves all rows from both tables, with NULLs where there is no match
SELECT College1.college_name, cdepartment1.name
FROM College1
FULL JOIN cdepartment1
ON College1.college_id = cdepartment1.college_id
ORDER BY College1.college_id;

-- 6. CROSS JOIN: Cartesian product (every row in College combined with every row in cdepartment)
```

COLLEGE_NAME	NAME
xyz	xyzp
abc	abc2
mno	mno3

3 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1 0.0.39 Copyright © 1999-2006, Oracle. All rights reserved.

Type here to search 14:19 07-10-2024

(7Z WhatsApp) SQL Commands

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
-- 5. Sub-query: Get the name of the college with maximum college_id
SELECT college_name, cdepartment.name
FROM College, cdepartment
WHERE College.college_id = cdepartment.college_id;
ORDER BY College.college_id;

-- 6. CROSS JOIN: Cartesian product (every row in College combined with every row in cdepartment)
SELECT College.college_name, cdepartment.name
FROM College, cdepartment
WHERE College.college_id = cdepartment.college_id;

-- 7. Sub-query: Retrieve the college with the maximum department_id
SELECT college_name
FROM College
WHERE college_id = (SELECT college_id FROM cdepartment WHERE department_id = (SELECT MAX(department_id) FROM cdepartment));

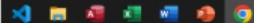
-- 8. Sub-query: Get the name of the department with minimum college_id
SELECT name
```

Results Explain Describe Saved SQL History

COLLEGE_NAME	NAME
xyz	xyzp
abc	abc2
mn0	mn03

3 rows returned in 0.02 seconds [CSV Export](#)

Language: en-us Application Express 2.1 0.00.39 Copyright © 1999-2006, Oracle. All rights reserved.

Type here to search  1420 07-10-2024

(7Z WhatsApp) SQL Commands

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
-- 5. Sub-query: Get the name of the college with maximum college_id
SELECT college_name, cdepartment.name
FROM College, cdepartment
WHERE College.college_id = cdepartment.college_id;
ORDER BY College.college_id;

-- 6. CROSS JOIN: Cartesian product (every row in College combined with every row in cdepartment)
SELECT College.college_name, cdepartment.name
FROM College, cdepartment
WHERE College.college_id = cdepartment.college_id;

-- 7. Sub-query: Retrieve the college with the maximum department_id
SELECT college_name
FROM College
WHERE college_id = (SELECT college_id FROM cdepartment WHERE department_id = (SELECT MAX(department_id) FROM cdepartment));

-- 8. Sub-query: Get the name of the department with minimum college_id
SELECT name
```

Results Explain Describe Saved SQL History

COLLEGE_NAME
mn0

1 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1 0.00.39 Copyright © 1999-2006, Oracle. All rights reserved.

Type here to search  1420 07-10-2024

(72) WhatsApp SQL Commands ORACLE Database Express Edition

User SYSTEM Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
WHERE college_id = (SELECT college_id FROM cdepartment1 WHERE department_id = (SELECT MAX(department_id) FROM cdepartment1));
-- 8. Sub-query: Get the name of the department with minimum college_id
SELECT name
FROM cdepartment1
WHERE college_id = (SELECT MIN(college_id) FROM College1);
-- 9. View: Create a view to show college name and department name
CREATE VIEW College_Department_View AS
SELECT College.college_name, cdepartment.name
FROM College
INNER JOIN cdepartment ON College.college_id = cdepartment.college_id;
-- 10. Query the view
```

Results Explain Describe Saved SQL History

NAME
xyzp

1 rows returned in 0.00 seconds CSV Export

Language en-us Application Express 2.1 0.0.39 Copyright © 1999-2006, Oracle. All rights reserved.

Type here to search 1421 07-10-2024

(72) WhatsApp SQL Commands ORACLE Database Express Edition

User SYSTEM Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
-- 8. Sub-query: Get the name of the department with minimum college_id
SELECT name
FROM cdepartment1
WHERE college_id = (SELECT MIN(college_id) FROM College1);
-- 9. View: Create a view to show college name and department name
CREATE VIEW College_Department_View AS
SELECT College.college_name, cdepartment1.name
FROM College1
INNER JOIN cdepartment1 ON College1.college_id = cdepartment1.college_id;
-- 10. Query the view
SELECT * FROM College_Department_View;
```

Results Explain Describe Saved SQL History

View created.

0.01 seconds

Language en-us Application Express 2.1 0.0.39 Copyright © 1999-2006, Oracle. All rights reserved.

Type here to search 1422 07-10-2024

(72) WhatsApp SQL Commands

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
-- 8. Sub-query: Get the name of the department with minimum college_id
SELECT name
FROM cdepartment1
WHERE college_id = (SELECT MIN(college_id) FROM College1);

-- 9. View: Create a view to show college name and department name
CREATE VIEW College_Department_View AS
SELECT College1.college_name, cdepartment1.name
FROM College1
INNER JOIN cdepartment1 ON College1.college_id = cdepartment1.college_id;

-- 10. Query the view
SELECT * FROM College_Department_View;
```

Results Explain Describe Saved SQL History

COLLEGE_NAME	NAME
xyz	xyzp
abc	abc2
mn0	mn03

3 rows returned in 0.00 seconds [CSV Export](#)

Language en-us Application Express 2.1 0.0.39 Copyright © 1999-2006, Oracle. All rights reserved.

Type here to search

Assignment 2.a

Input:-

Assignment 2.b

Input:-

```
CREATE TABLE Employee (
    emp_id NUMBER(5) PRIMARY KEY,
    emp_name VARCHAR2(50),
    dept_id NUMBER(5),
    salary NUMBER(10)
);

CREATE TABLE Department (
    dept_id NUMBER(5) PRIMARY KEY,
    dept_name VARCHAR2(50)
);

INSERT INTO Employee (emp_id, emp_name, dept_id, salary)
VALUES (1001, 'Alice', 101, 60000);

INSERT INTO Employee (emp_id, emp_name, dept_id, salary)
VALUES (1002, 'Bob', 102, 45000);

INSERT INTO Employee (emp_id, emp_name, dept_id, salary)
VALUES (1003, 'Charlie', 103, 70000);

INSERT INTO Department (dept_id, dept_name)
VALUES (101, 'HR');

INSERT INTO Department (dept_id, dept_name)
VALUES (102, 'Finance');

INSERT INTO Department (dept_id, dept_name)
VALUES (103, 'IT');

-- 1. Select Employees with Salary Greater Than 50000
SELECT emp_name, salary
FROM Employee
WHERE salary > 50000;
```

-- 2. Select Employees with Salary Between 30000 and 80000

```
SELECT emp_name, salary  
FROM Employee  
WHERE salary BETWEEN 30000 AND 80000;
```

-- 3. Update Employee's Salary with a Condition

```
UPDATE Employee  
SET salary = salary * 1.1 -- 10% increase
```

```
WHERE emp_name = 'Alice';  
-- 4. Delete Employees with Salary Less Than 30000
```

```
DELETE FROM Employee  
WHERE salary < 30000;  
-- 5. Select Employees and Departments (Using JOIN)
```

```
SELECT e.emp_name, d.dept_name  
FROM Employee e  
JOIN Department d  
ON e.dept_id = d.dept_id;
```

-- 6. Select Employees Using IN Operator

```
SELECT emp_name, salary  
FROM Employee  
WHERE dept_id IN (101, 102);
```

-- 7. Select Maximum Salary and Minimum Salary

```
SELECT MAX(salary) AS max_salary, MIN(salary) AS min_salary  
FROM Employee;
```

-- 8. Union Query: Combine Employees from Two Departments

```
SELECT emp_name, dept_id  
FROM Employee  
WHERE dept_id = 101  
UNION  
SELECT emp_name, dept_id  
FROM Employee  
WHERE dept_id = 102;
```

-- 9. Count the Number of Employees in Each Department

```
SELECT dept_id, COUNT(*) AS num_employees
```

```
FROM Employee
```

```
GROUP BY dept_id;
```

-- 10. Select Employees Using LIKE (Pattern Matching)

```
SELECT emp_name, salary
```

```
FROM Employee
```

```
WHERE emp_name LIKE 'A%';
```

Output:-

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The user is connected as SYSTEM. The SQL editor contains the following code:

```
-- Create Employee Table
CREATE TABLE Employee1 (
    emp_id NUMBER(5) PRIMARY KEY,
    emp_name VARCHAR2(50),
    dept_id NUMBER(5),
    salary NUMBER(10)
);

-- Create Department Table
CREATE TABLE Department (
    dept_id NUMBER(5) PRIMARY KEY,
    dept_name VARCHAR2(50)
);
```

The results pane shows the message "Table created." and "0.00 seconds". The status bar at the bottom right indicates "Application Express 2.1.0.00.39" and "Copyright © 1999, 2006, Oracle. All rights reserved." The taskbar at the bottom shows various application icons.

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The user is connected as SYSTEM. The SQL editor contains the following code:

```
salary NUMBER(10)
);

-- Create Department Table
CREATE TABLE Department1 (
    dept_id NUMBER(5) PRIMARY KEY,
    dept_name VARCHAR2(50)
);

-- Insert Data into Employee Table
INSERT INTO Employee (emp_id, emp_name, dept_id, salary)
VALUES (1001, 'Alice', 101, 60000);

INSERT INTO Employee (emp_id, emp_name, dept_id, salary)
```

The results pane shows the message "Table created." and "0.00 seconds". The status bar at the bottom right indicates "Application Express 2.1.0.00.39" and "Copyright © 1999, 2006, Oracle. All rights reserved." The taskbar at the bottom shows various application icons.

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The user is connected as SYSTEM. The SQL editor contains the following code:

```
salary NUMBER(10)
);

-- Create Department Table
CREATE TABLE Department1 (
    dept_id NUMBER(5) PRIMARY KEY,
    dept_name VARCHAR2(50)
);

-- Insert Data into Employee Table
INSERT INTO Employee (emp_id, emp_name, dept_id, salary)
VALUES (1001, 'Alice', 101, 60000);

INSERT INTO Employee (emp_id, emp_name, dept_id, salary)
```

The results pane shows the message "Table created." and "0.00 seconds". The status bar at the bottom right indicates "Application Express 2.1.0.00.39" and "Copyright © 1999, 2006, Oracle. All rights reserved." The taskbar at the bottom shows various application icons.

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
dept_id NUMBER(3) PRIMARY KEY,
dept_name VARCHAR2(50)
);

-- Insert Data into Employee Table
INSERT INTO Employee1 (emp_id, emp_name, dept_id, salary)
VALUES (1001, 'Alice', 101, 60000);

INSERT INTO Employee (emp_id, emp_name, dept_id, salary)
VALUES (1002, 'Bob', 102, 45000);

INSERT INTO Employee (emp_id, emp_name, dept_id, salary)
VALUES (1003, 'Charlie', 103, 70000);

-- Insert Data into Department Table
```

The results section shows the message: "1 row(s) inserted." and "0.00 seconds".

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is identical to the one in the previous screenshot:

```
dept_id NUMBER(3) PRIMARY KEY,
dept_name VARCHAR2(50)
);

-- Insert Data into Employee Table
INSERT INTO Employee1 (emp_id, emp_name, dept_id, salary)
VALUES (1001, 'Alice', 101, 60000);

INSERT INTO Employee (emp_id, emp_name, dept_id, salary)
VALUES (1002, 'Bob', 102, 45000);

INSERT INTO Employee (emp_id, emp_name, dept_id, salary)
VALUES (1003, 'Charlie', 103, 70000);

-- Insert Data into Department Table
```

The results section shows the message: "1 row(s) inserted." and "0.00 seconds".

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is identical to the ones above:

```
dept_id NUMBER(3) PRIMARY KEY,
dept_name VARCHAR2(50)
);

-- Insert Data into Employee Table
INSERT INTO Employee1 (emp_id, emp_name, dept_id, salary)
VALUES (1001, 'Alice', 101, 60000);

INSERT INTO Employee (emp_id, emp_name, dept_id, salary)
VALUES (1002, 'Bob', 102, 45000);

INSERT INTO Employee (emp_id, emp_name, dept_id, salary)
VALUES (1003, 'Charlie', 103, 70000);

-- Insert Data into Department Table
```

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO Employee1 (emp_id, emp_name, dept_id, salary)
VALUES (1003, 'Charlie', 103, 70000);

-- Insert Data into Department Table
INSERT INTO Department1 (dept_id, dept_name)
VALUES (101, 'HR');

INSERT INTO Department (dept_id, dept_name)
VALUES (102, 'Finance');

INSERT INTO Department (dept_id, dept_name)
VALUES (103, 'IT');

-- 1. Select Employees with Salary Greater Than 50000
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.00 seconds

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2010, Oracle. All rights reserved.

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO Employee1 (emp_id, emp_name, dept_id, salary)
VALUES (1003, 'Charlie', 103, 70000);

-- Insert Data into Department Table
INSERT INTO Department1 (dept_id, dept_name)
VALUES (101, 'HR');

INSERT INTO Department1 (dept_id, dept_name)
VALUES (102, 'Finance');

INSERT INTO Department (dept_id, dept_name)
VALUES (103, 'IT');

-- 1. Select Employees with Salary Greater Than 50000
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.00 seconds

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.

(7) WhatsApp SQL Commands

User SYSTEM

ORACLE Database Express Edition

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
INSERT INTO department VALUES (103, 'IT');
-- 1. Select Employees with Salary Greater Than 50000
SELECT emp_name, salary
FROM Employee
WHERE salary > 50000;
-- 2. Select Employees with Salary Between 30000 and 80000
SELECT emp_name, salary
FROM Employee
WHERE salary BETWEEN 30000 AND 80000;
-- 3. Update Employee's Salary with a Condition
UPDATE Employee
```

Results Explain Describe Saved SQL History

EMP_NAME	SALARY
Alice	60000
Charlie	70000

2 rows returned in 0.00 seconds CSV Export

Language en-us Application Express 2.1.0.0.39 Copyright © 1999-2010, Oracle. All rights reserved.

Type here to search 14:47 07-10-2014

(7) WhatsApp SQL Commands

User SYSTEM

ORACLE Database Express Edition

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
INSERT INTO department VALUES (103, 'IT');
-- 1. Select Employees with Salary Greater Than 50000
SELECT emp_name, salary
FROM Employee
WHERE salary > 50000;
-- 2. Select Employees with Salary Between 30000 and 80000
SELECT emp_name, salary
FROM Employee
WHERE salary BETWEEN 30000 AND 80000;
-- 3. Update Employee's Salary with a Condition
UPDATE Employee
```

Results Explain Describe Saved SQL History

EMP_NAME	SALARY
Alice	60000
Bob	45000
Charlie	70000

3 rows returned in 0.00 seconds CSV Export

Language en-us Application Express 2.1.0.0.39 Copyright © 1999-2010, Oracle. All rights reserved.

Type here to search 14:47 07-10-2014

ORACLE® Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

```
WHERE salary BETWEEN 30000 AND 80000;
-- 3. Update Employee's Salary with a Condition
UPDATE Employee
SET salary = salary * 1.1 -- 10% increase
WHERE emp_name = 'Alice';

-- 4. Delete Employees with Salary Less Than 30000
DELETE FROM Employee
WHERE salary < 30000;

-- 5. Select Employees and Departments (Using JOIN)
SELECT e.emp_name, d.dept_name
FROM Employee e
```

Results Explain Describe Saved SQL History

1 row(s) updated.

0.00 seconds

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.

ORACLE® Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

```
WHERE salary BETWEEN 30000 AND 80000;
-- 3. Update Employee's Salary with a Condition
UPDATE Employee
SET salary = salary * 1.1 -- 10% increase
WHERE emp_name = 'Alice';

-- 4. Delete Employees with Salary Less Than 30000
DELETE FROM Employee
WHERE salary < 30000;

-- 5. Select Employees and Departments (Using JOIN)
SELECT e.emp_name, d.dept_name
FROM Employee e
```

Results Explain Describe Saved SQL History

0 row(s) deleted.

0.00 seconds

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
-- 5. Select Employees and Departments (Using JOIN)
SELECT e.emp_name, d.dept_name
FROM Employee e
JOIN Department d
ON e.dept_id = d.dept_id;

-- 6. Select Employees Using IN Operator
SELECT emp_name, salary
FROM Employee
WHERE dept_id IN (101, 102);

-- 7. Select Maximum Salary and Minimum Salary
SELECT MAX(salary) AS max_salary, MIN(salary) AS min_salary
FROM Employee;
```

Results Explain Describe Saved SQL History

EMP_NAME	DEPT_NAME
Alice	HR
Bob	Finance
Charlie	IT

3 rows returned in 0.01 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.

Type here to search 14:48 07-10-2024

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit

```
-- 6. Select Employees Using IN Operator
SELECT emp_name, salary
FROM Employee
WHERE dept_id IN (101, 102);

-- 7. Select Maximum Salary and Minimum Salary
SELECT MAX(salary) AS max_salary, MIN(salary) AS min_salary
FROM Employee;

-- 8. Union Query: Combine Employees from Two Departments
SELECT emp_name, dept_id
FROM Employee
WHERE dept_id = 101
```

Results Explain Describe Saved SQL History

EMP_NAME	SALARY
Alice	66000
Bob	45000

2 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2016, Oracle. All rights reserved.

Type here to search 14:49 07-10-2024

(7) WhatsApp SQL Commands 127.0.0.1:8080/apex/f?p=4500:1003:1783013766803998::NO::

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit Save Run

```
-- 6. Select Employees Using IN Operator
SELECT emp_name, salary
FROM Employee
WHERE dept_id IN (101, 102);

-- 7. Select Maximum Salary and Minimum Salary
SELECT MAX(salary) AS max_salary, MIN(salary) AS min_salary
FROM Employee;

-- 8. Union Query: Combine Employees from Two Departments
SELECT emp_name, dept_id
FROM Employee
WHERE dept_id = 101;
```

Results Explain Describe Saved SQL History

MAX_SALARY	MIN_SALARY
70000	45000

1 rows returned in 0.00 seconds

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2010, Oracle. All rights reserved.

Type here to search 14:49 07-10-2024

(7) WhatsApp SQL Commands 127.0.0.1:8080/apex/f?p=4500:1003:1783013766803998::NO::

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit Save Run

```
-- 3. Union Query: Combine Employees from Two Departments
SELECT emp_name, dept_id
FROM Employee
WHERE dept_id = 101
UNION
SELECT emp_name, dept_id
FROM Employee
WHERE dept_id = 102;

-- 9. Count the Number of Employees in Each Department
SELECT dept_id, COUNT(*) AS num_employees
FROM Employee
GROUP BY dept_id;

-- 10. Select Employees Using LIKE / Pattern Matching
```

Results Explain Describe Saved SQL History

EMP_NAME	DEPT_ID
Alice	101
Bob	102

2 rows returned in 0.00 seconds

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

Type here to search 14:50 07-10-2024

(7) WhatsApp SQL Commands

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit Display | 10 Save Run

```
UNION
SELECT emp_name, dept_id
FROM Employee1
WHERE dept_id = 102;

-- 9. Count the Number of Employees in Each Department
SELECT dept_id, COUNT(*) AS num_employees
FROM Employee1
GROUP BY dept_id;

-- 10. Select Employees Using LIKE (Pattern Matching)
SELECT emp_name, salary
FROM Employee1
WHERE emp_name LIKE 'A%';
```

Results Explain Describe Saved SQL History

DEPT_ID	NUM_EMPLOYEES
102	1
101	1
103	1

3 rows returned in 0.02 seconds CSV Export

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999-2010, Oracle. All rights reserved.

Type here to search 14:50 07-10-2014

(7) WhatsApp SQL Commands

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit Display | 10 Save Run

```
UNION
SELECT emp_name, dept_id
FROM Employee1
WHERE dept_id = 102;

-- 9. Count the Number of Employees in Each Department
SELECT dept_id, COUNT(*) AS num_employees
FROM Employee1
GROUP BY dept_id;

-- 10. Select Employees Using LIKE (Pattern Matching)
SELECT emp_name, salary
FROM Employee1
WHERE emp_name LIKE 'A%';
```

Results Explain Describe Saved SQL History

EMP_NAME	SALARY
Alice	66000

1 rows returned in 0.00 seconds CSV Export

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999-2010, Oracle. All rights reserved.

Type here to search 14:50 07-10-2014