QUESTION 1

IDENTIFY RELATIONS

MANAGES = 1:1 relationship between the department and the employee.

Participation by employees is limited.

HAS = Relationship type of DEPARTMENT:EMPLOYEE is 1:N. Both contributions are complete.

CONTROLS = Relationship type of DEPARTMENT:PROJECT is 1:N.

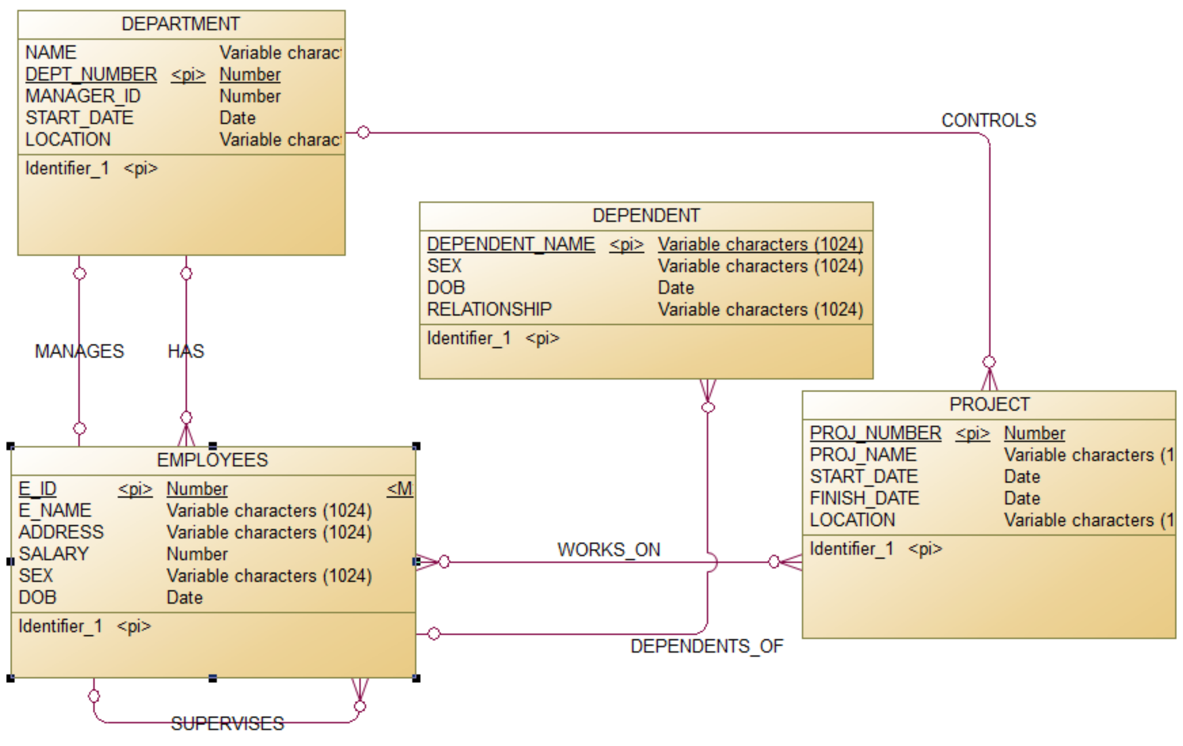
While DEPARTMENT only has a limited involvement, PROJECT has full engagement. After consulting with the users, it has been judged to be partial, indicating that Some departments might have no project control.

SUPERVISES= Relationship type 1:N between employee and employee who is acting as a supervisor (in the supervisee role). The two contributions are ,it is discovered to be partial once users point out that not every employee is a Not every employee has a supervisor, though.

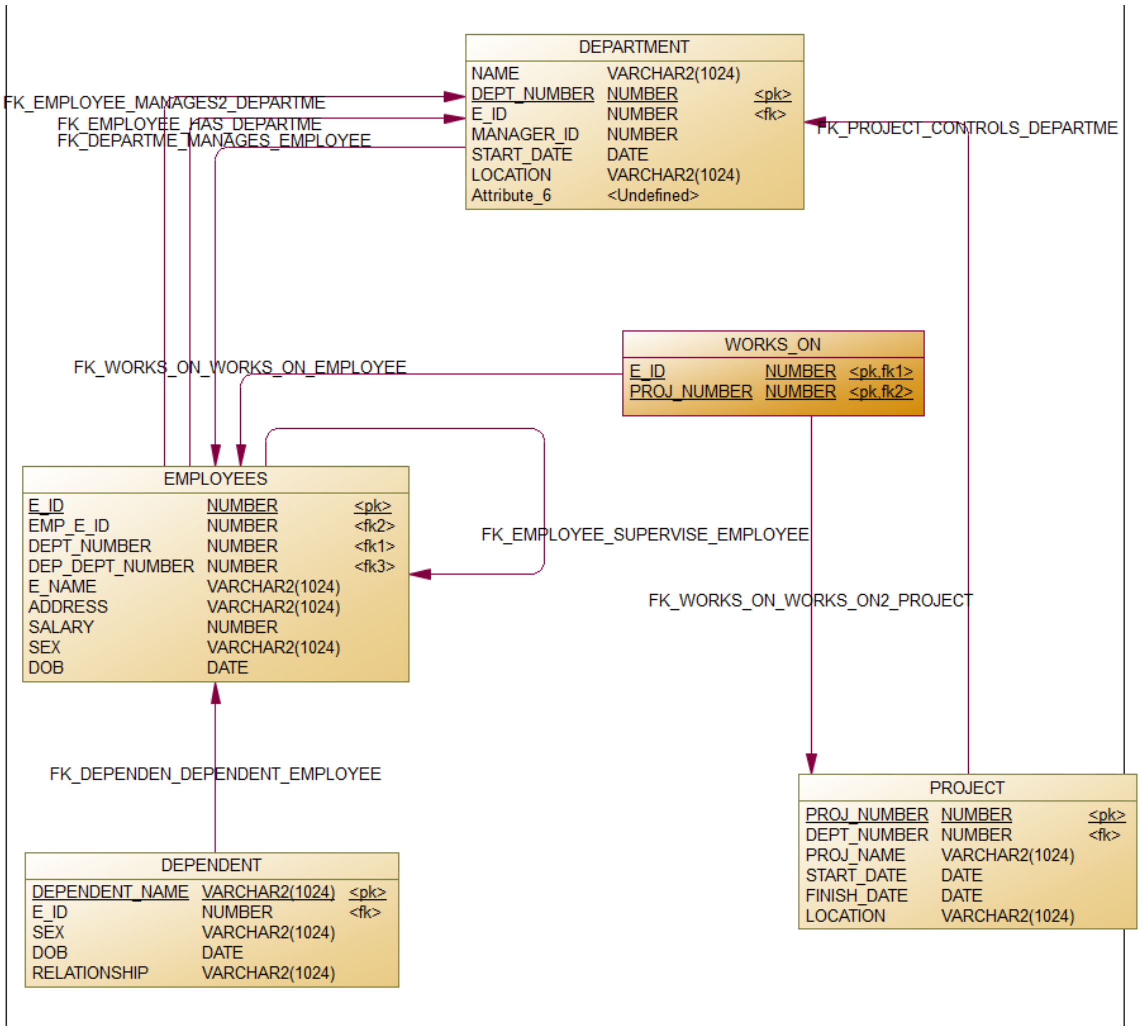
After users indicate that a project can have several employees, WORKS_ON is found to be a M:N

DEPENDENT_OF = 1:1, is a type. Partial employee participation compared to full employee participation

CDM



PDM



QUESTION 2

SQL FOR TABLES

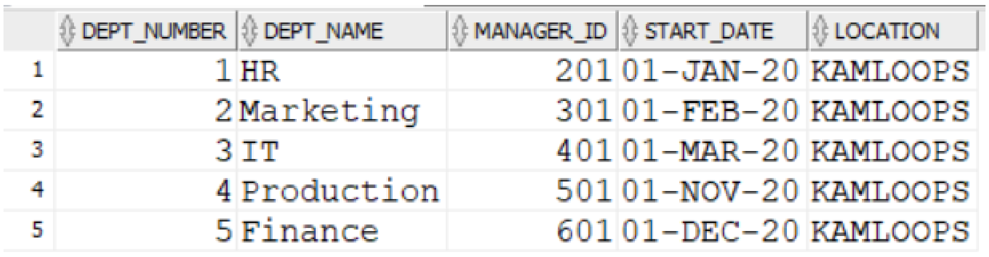
```
CREATE TABLE DEPARTMENT (
  DEPT_NUMBER INTEGER PRIMARY KEY,
  DEPT_NAME VARCHAR(30) check(DEPT_NAME in('HR','Marketing', 'IT', 'Production', 'Finance')) ,
  MANAGER_ID INTEGER ,
  START_DATE DATE ,
  LOCATION varchar(30)
);
```

```
CREATE TABLE PROJECT (
   PROJ_NUMBER INTEGER PRIMARY KEY,
   PROJ_NAME VARCHAR(30) NOT NULL,
   START_DATE DATE NOT NULL,
   FINISH_DATE DATE NOT NULL,
   LOCATION VARCHAR(30) NOT NULL,
   DEPT_NUMBER INTEGER NOT NULL,
   FOREIGN KEY (DEPT_NUMBER) REFERENCES DEPARTMENT (DEPT_NUMBER)
);

CREATE TABLE EMPLOYEE (
   E_ID INTEGER PRIMARY KEY,
   E_NAME VARCHAR(30) NOT NULL,
   ADDRESS VARCHAR(100) NOT NULL,
   SALARY DECIMAL(10,2) NOT NULL,
   SEX CHAR(1) NOT NULL,
   DOB DATE NOT NULL ,
   DEPT_NUMBER INTEGER NOT NULL,
   MANAGER_ID INTEGER NOT NULL,
   FOREIGN KEY (DEPT_NUMBER) REFERENCES DEPARTMENT (DEPT_NUMBER),
   FOREIGN KEY (MANAGER_ID) REFERENCES EMPLOYEE (E_ID)
);

CREATE TABLE DEPENDENT (
   DEPENDENT_ID INTEGER PRIMARY KEY,
   DEPENDENT_NAME VARCHAR(30) NOT NULL,
   SEX CHAR(1) NOT NULL,
   DOB DATE NOT NULL,
   RELATIONSHIP VARCHAR(20) NOT NULL,
   E_ID INTEGER NOT NULL,
    FOREIGN KEY (E_ID) REFERENCES EMPLOYEE (E_ID)
);
```

QUESTION 3

//INSERTION SQL CODE

```
INSERT INTO Department
VALUES (1, 'HR', 201, '01-JAN-2020','KAMLOOPS');
INSERT INTO Department
VALUES (2, 'Marketing', 301, '01-FEB-2020','KAMLOOPS');
INSERT INTO Department
VALUES (3, 'IT', 401, '01-MAR-2020','KAMLOOPS');
INSERT INTO Department
VALUES (4, 'Production', 501, '01-NOV-2020','KAMLOOPS');
INSERT INTO Department
VALUES (5, 'Finance', 601, '01-DEC-2020','KAMLOOPS');
```



	DEPT_NUMBER	DEPT_NAME	MANAGER_ID	START_DATE	LOCATION
1	1	HR	201	01-JAN-20	KAMLOOPS
2	2	Marketing	301	01-FEB-20	KAMLOOPS
3	3	IT	401	01-MAR-20	KAMLOOPS
4	4	Production	501	01-NOV-20	KAMLOOPS
5	5	Finance	601	01-DEC-20	KAMLOOPS

```
INSERT INTO EMPLOYEE (E_ID, E_NAME, ADDRESS, SALARY, SEX, DOB, DEPT_NUMBER,
MANAGER_ID)
VALUES (201, 'John Doe', '123 Main St', 50000, 'M', TO_DATE('1990-01-01', 'YYYY-MM-DD'), 1, 201);

INSERT INTO EMPLOYEE (E_ID, E_NAME, ADDRESS, SALARY, SEX, DOB, DEPT_NUMBER,
MANAGER_ID)
VALUES (301, 'Jane Smith', '456 Maple Ave', 60000, 'F', TO_DATE('1985-05-15', 'YYYY-MM-DD'), 2,
301);
```

INSERT INTO EMPLOYEE (E_ID, E_NAME, ADDRESS, SALARY, SEX, DOB, DEPT_NUMBER, MANAGER_ID)

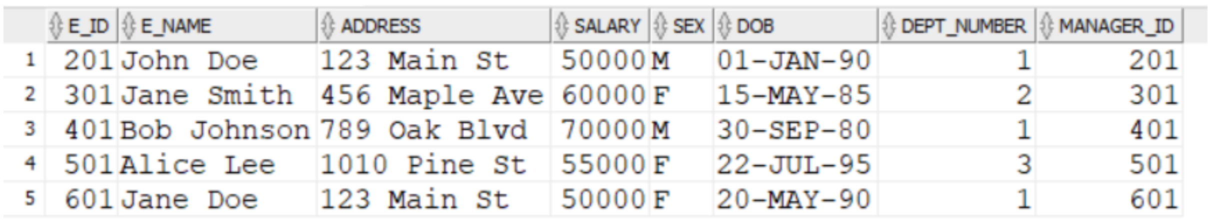
VALUES (401, 'Bob Johnson', '789 Oak Blvd', 70000, 'M', TO_DATE('1980-09-30', 'YYYY-MM-DD'), 1, 401);

INSERT INTO EMPLOYEE (E_ID, E_NAME, ADDRESS, SALARY, SEX, DOB, DEPT_NUMBER, MANAGER_ID)

VALUES (501, 'Alice Lee', '1010 Pine St', 55000, 'F', TO_DATE('1995-07-22', 'YYYY-MM-DD'), 3, 501);

INSERT INTO EMPLOYEE (E_ID, E_NAME, ADDRESS, SALARY, SEX, DOB, DEPT_NUMBER, MANAGER_ID)

VALUES (601, 'Jane Doe', '123 Main St', 50000.00, 'F', TO_DATE('1990-05-20', 'YYYY-MM-DD'), 1, 601);



	E_ID	E_NAME	ADDRESS	SALARY	SEX	DOB	DEPT_NUMBER	MANAGER_ID
1	201	John Doe	123 Main St	50000	M	01-JAN-90	1	201
2	301	Jane Smith	456 Maple Ave	60000	F	15-MAY-85	2	301
3	401	Bob Johnson	789 Oak Blvd	70000	M	30-SEP-80	1	401
4	501	Alice Lee	1010 Pine St	55000	F	22-JUL-95	3	501
5	601	Jane Doe	123 Main St	50000	F	20-MAY-90	1	601

INSERT INTO PROJECT (PROJ_NUMBER, PROJ_NAME, START_DATE, FINISH_DATE, LOCATION, DEPT_NUMBER)

VALUES

(1, 'Project A', TO_DATE('2022-01-01', 'YYYY-MM-DD'), TO_DATE('2022-06-30', 'YYYY-MM-DD'), 'New York', 1);

INSERT INTO PROJECT (PROJ_NUMBER, PROJ_NAME, START_DATE, FINISH_DATE, LOCATION, DEPT_NUMBER)

VALUES

(2, 'Project B', TO_DATE('2022-02-01', 'YYYY-MM-DD'), TO_DATE('2022-09-30', 'YYYY-MM-DD'), 'London', 2);

INSERT INTO PROJECT (PROJ_NUMBER, PROJ_NAME, START_DATE, FINISH_DATE, LOCATION, DEPT_NUMBER)

VALUES

(3, 'Project C', TO_DATE('2022-03-01', 'YYYY-MM-DD'), TO_DATE('2022-08-31', 'YYYY-MM-DD'), 'Sydney', 3);

INSERT INTO PROJECT (PROJ_NUMBER, PROJ_NAME, START_DATE, FINISH_DATE, LOCATION, DEPT_NUMBER)

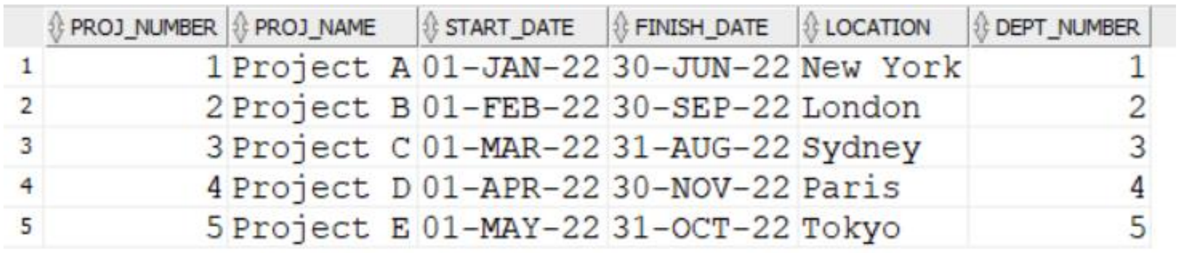
VALUES

(4, 'Project D', TO_DATE('2022-04-01', 'YYYY-MM-DD'), TO_DATE('2022-11-30', 'YYYY-MM-DD'), 'Paris', 4);

INSERT INTO PROJECT (PROJ_NUMBER, PROJ_NAME, START_DATE, FINISH_DATE, LOCATION, DEPT_NUMBER)

VALUES

(5, 'Project E', TO_DATE('2022-05-01', 'YYYY-MM-DD'), TO_DATE('2022-10-31', 'YYYY-MM-DD'), 'Tokyo', 5);



	PROJ_NUMBER	PROJ_NAME	START_DATE	FINISH_DATE	LOCATION	DEPT_NUMBER
1	1	Project A	01-JAN-22	30-JUN-22	New York	1
2	2	Project B	01-FEB-22	30-SEP-22	London	2
3	3	Project C	01-MAR-22	31-AUG-22	Sydney	3
4	4	Project D	01-APR-22	30-NOV-22	Paris	4
5	5	Project E	01-MAY-22	31-OCT-22	Tokyo	5

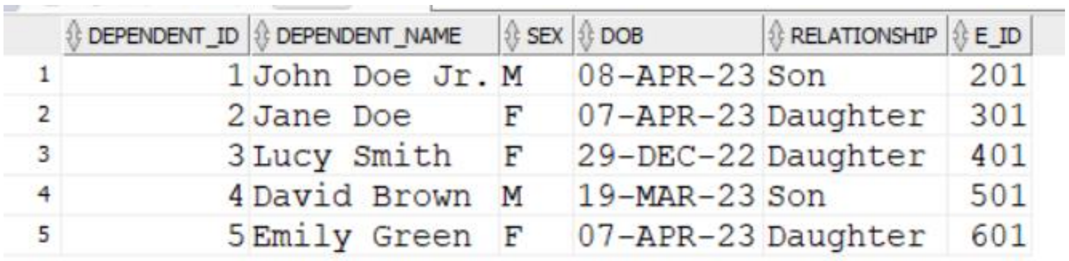
INSERT INTO DEPENDENT VALUES (1, 'John Doe Jr.', 'M', sysdate , 'Son', 201);

INSERT INTO DEPENDENT (DEPENDENT_ID, DEPENDENT_NAME, SEX, DOB, RELATIONSHIP, E_ID) VALUES (2, 'Jane Doe', 'F', sysdate-1, 'Daughter', 301);

INSERT INTO DEPENDENT (DEPENDENT_ID, DEPENDENT_NAME, SEX, DOB, RELATIONSHIP, E_ID) VALUES (3, 'Lucy Smith', 'F', sysdate-100, 'Daughter', 401);

INSERT INTO DEPENDENT (DEPENDENT_ID, DEPENDENT_NAME, SEX, DOB, RELATIONSHIP, E_ID) VALUES (4, 'David Brown', 'M', sysdate-20, 'Son', 501);

INSERT INTO DEPENDENT (DEPENDENT_ID, DEPENDENT_NAME, SEX, DOB, RELATIONSHIP, E_ID) VALUES (5, 'Emily Green', 'F', sysdate-1, 'Daughter', 601);



	DEPENDENT_ID	DEPENDENT_NAME	SEX	DOB	RELATIONSHIP	E_ID
1	1	John Doe Jr.	M	08-APR-23	Son	201
2	2	Jane Doe	F	07-APR-23	Daughter	301
3	3	Lucy Smith	F	29-DEC-22	Daughter	401
4	4	David Brown	M	19-MAR-23	Son	501
5	5	Emily Green	F	07-APR-23	Daughter	601

QUESTION 4

a)

```
SELECT EMPLOYEE.E_NAME, PROJECT.PROJ_NAME

FROM EMPLOYEE

JOIN PROJECT ON EMPLOYEE.DEPT_NUMBER = PROJECT.DEPT_NUMBER;
```

E_NAME	PROJ_NAME

b)

```
SELECT EMPLOYEE.E_NAME AS MANAGER_NAME, DEPARTMENT.DEPT_NAME,
DEPARTMENT.START_DATE

FROM EMPLOYEE

JOIN DEPARTMENT ON EMPLOYEE.E_ID = DEPARTMENT.MANAGER_ID;
```

MANAGER...	DEPT_NAME	START_D...

c)

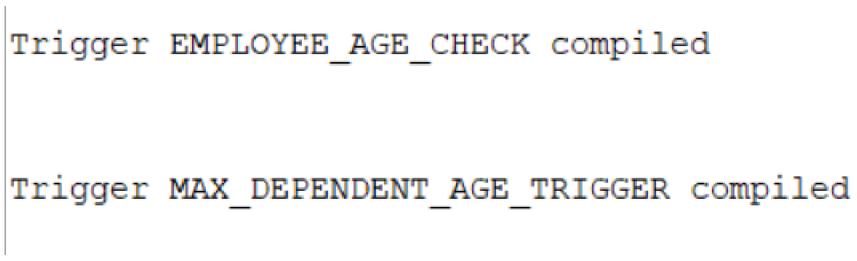
```
SELECT EMPLOYEE.E_NAME AS EMPLOYEE_NAME, EMPLOYEE.DOB AS EMPLOYEE_DOB,
DEPENDENT.DEPENDENT_NAME, DEPENDENT.DOB AS DEPENDENT_DOB

FROM EMPLOYEE

JOIN DEPENDENT ON EMPLOYEE.E_ID = DEPENDENT.E_ID

WHERE MONTH(EMPLOYEE.DOB) = 12 OR MONTH(DEPENDENT.DOB) = 12;
```

PROCEDURE/FUNCTIONS/TRIGGERS TO IMPLEMENT CONSTRAINTS

TRIGGER TO IMPLEMENT 16 YEARS OF AGE IN EMPLOYEES

```
CREATE OR REPLACE TRIGGER EMPLOYEE_AGE_CHECK

BEFORE INSERT OR UPDATE ON EMPLOYEE

FOR EACH ROW

DECLARE

  EMP_AGE NUMBER(3);
```

```
BEGIN
   EMP_AGE := MONTHS_BETWEEN(SYSDATE, :NEW.DOB) / 12;
   IF EMP_AGE < 16 THEN
      RAISE_APPLICATION_ERROR(-20001, 'EMPLOYEES MUST BE AT LEAST 16 YEARS OLD.');
   END IF;
END;



CREATE OR REPLACE TRIGGER MAX_DEPENDENT_AGE_TRIGGER
BEFORE INSERT OR UPDATE ON DEPENDENT
FOR EACH ROW
DECLARE
   AGE NUMBER(3);
BEGIN
   -- CHECK IF THE DEPENDENT IS A SPOUSE, IF YES, ALLOW THE INSERT/UPDATE
   IF :NEW.RELATIONSHIP = 'SPOUSE' THEN
      RETURN;
   END IF;

   -- CALCULATE THE AGE OF THE DEPENDENT
   AGE := TRUNC(MONTHS_BETWEEN(SYSDATE, :NEW.DOB) / 12);

   -- CHECK IF THE AGE IS GREATER THAN 16, IF YES, DISALLOW THE INSERT/UPDATE
   IF AGE > 16 THEN
      RAISE_APPLICATION_ERROR(-20001, 'EXCEPT FOR SPOUSES, DEPENDENTS MUST BE UNDER 
OR EQUAL 16 YEARS OLD..');
   END IF;
END;
```



```
Trigger EMPLOYEE_AGE_CHECK compiled


Trigger MAX_DEPENDENT_AGE_TRIGGER compiled
```