**CS4347.001 - Database Systems**

**Instructor: Nidhiben Solanki**

**Project Report**

ABC Company Project

Group 6

Aroofa Mohammad

Alice Huong Nguyen

Muskan Vadsaria

Document the final term project report. (15%)

**a) Problem description**

Candidate\_ID → can receive from either Employee\_ID or PEmployee\_ID

So Created a Candidate entity with attributes Candidate\_ID(primary key), Employee\_ID (foreign key, PEmpolyee\_ID(foreign key))

**b) Individual contribution breakdown**

Aroofa Mohammad: EER Diagram (beginning parts), Normalization, Relational Schema.

Alice Nguyen: EER Diagram (ending parts), Database schema, and SQL statement to create database, table and other structure. SQL statement to create views and answer all questions.

Muskan Vadsaria: Dependency diagram.

**c) Project questions (Answer questions listed in this project)**

Normalize all of your tables to the third normal form. Make any necessary changes to the EER. Explain why these changes needed to be made.

A change was needed in the interview entity because the Candidate\_ID can receive both the Employee\_ID and PotentialEmployee\_ID

**d) EER diagram with all assumptions.**

Assumption: Person has to be from the three Customer, Employee, and Potential Employee.

Diagram

Description automatically generated

**e) Relation schema after normalization. All relations must be in 3NF. The relation schema should include primary keys as well as foreign keys (if any) for all relations.**

**Table

Description automatically generated with medium confidence**

**f) All requested SQL statements**

**CREATE TABLE EMPLOYEE**

|  |  |  |
| --- | --- | --- |
| **( Lname** | **VARCHAR (15)** | **NOT NULL,** |
| **Fname** | **VARCHAR (15)** | **NOT NULL,** |
| **Ssn** | **CHAR (9)** | **NOT NULL,** |
| **Bdate** | **DATE,** |  |
| **Address** | **VARCHAR (30),** |  |
| **Phone** | **CHAR (10),** |  |
| **Gender** | **CHAR,** |  |
| **Rank** | **VARCHAR (15),** |  |
| **Title** | **VARCHAR (15),** |  |
| **Super\_ssn** | **CHAR (9),** | **DEFAULT ‘123456789’** |
| **Dno** | **INT** | **NOT NULL           DEFAULT 1** |
| **PRIMARY KEY (Ssn),** | | |
| **FOREIGN KEY (Super\_ssn) REFERENCES EMPLOYEE (Ssn)** | | |
| **ON DELETE SET NULL                        ON UPDATE CASCADE,** | | |
| **FOREIGN KEY (Dno) REFERENCES DEPARTMENT (Dnumber)** | | |
| **ON DELETE SET NULL          ON UPDATE CASCADE);** | | |

**CREATE TABLE POTENTIAL\_EMPLOYEE**

|  |  |  |
| --- | --- | --- |
| **( Lname** | **VARCHAR (15)** | **NOT NULL,** |
| **Fname** | **VARCHAR (15)** | **NOT NULL,** |
| **Pssn** | **CHAR (9)** | **NOT NULL,** |
| **Bdate** | **DATE,** |  |
| **Address** | **VARCHAR (30),** |  |
| **Phone** | **CHAR (10),** |  |
| **Gender** | **CHAR,** |  |
| **PRIMARY KEY (Pssn));** | | |

**CREATE TABLE CUSTOMER**

|  |  |  |
| --- | --- | --- |
| **( Lname** | **VARCHAR (15)** | **NOT NULL,** |
| **Fname** | **VARCHAR (15)** | **NOT NULL,** |
| **Cssn** | **CHAR (9)** | **NOT NULL,** |
| **Bdate** | **DATE,** |  |
| **Address** | **VARCHAR (30),** |  |
| **Phone** | **CHAR (10),** |  |
| **Gender** | **CHAR,** |  |
| **Salesmen** | **VARCHAR (15),** |  |
| **PRIMARY KEY (Cssn));** | | |

**CREATE TABLE MONTHLY\_SALARY**

|  |  |  |
| --- | --- | --- |
| **( Essn** | **CHAR (9)** | **NOT NULL,** |
| **Transaction\_number** | **INT,** |  |
| **Pay\_date** | **DATE,** |  |
| **Amount** | **DECIMAL (10,2),** |  |
| **PRIMARY KEY (Essn, Transaction\_number),** | | |
| **UNIQUE (Transaction\_number),** | | |
| **FOREIGN KEY (Essn) REFERENCES EMPLOYEE (Ssn)** | | |
| **ON DELETE CASCADE                        ON UPDATE CASCADE);** | | |

**CREATE TABLE DEPARTMENT**

|  |  |  |
| --- | --- | --- |
| **( Dname** | **VARCHAR (15)** | **NOT NULL,** |
| **Dnumber** | **INT** | **NOT NULL,** |
| **Mgr\_ssn** | **CHAR (9),** |  |
| **PRIMARY KEY (Dnumber),** | | |
| **FOREIGN KEY (Mgr\_ssn) REFERENCES EMPLOYEE (Ssn)** | | |
| **ON DELETE SET NULL                  ON UPDATE CASCADE);** | | |

**CREATE TABLE WORK\_FOR**

|  |  |  |
| --- | --- | --- |
| **( Emp\_ID** | **CHAR (9)** | **NOT NULL,** |
| **Dnum** | **INT** | **NOT NULL,** |
| **Shift\_Start** | **VARCHAR (10),** |  |
| **Shift\_End** | **VARCHAR (10),** |  |
| **PRIMARY KEY (Emp\_ID, Dnum),** | | |
| **FOREIGN KEY (Emp\_ID) REFERENCES EMPLOYEE (Ssn)** | | |
| **ON DELETE CASCADE                     ON UPDATE CASCADE,** | | |
| **FOREIGN KEY (Dnum) REFERENCES DEPARTMENT (Dnumber)** | | |
| **ON DELETE CASCADE             ON UPDATE CASCADE);** | | |

**CREATE TABLE JOB**

|  |  |  |
| --- | --- | --- |
| **( Job\_ID** | **INT** | **NOT NULL,** |
| **Job\_description** | **VARCHAR (50),** |  |
| **Job\_date** | **DATE,** |  |
| **Dept\_ID** | **INT** | **NOT NULL,** |
| **PRIMARY KEY (Job\_ID),** | | |
| **FOREIGN KEY (Dept\_ID) REFERENCES DEPARTMENT (Dnumber)** | | |
| **ON DELETE CASCADE           ON UPDATE CASCADE);** | | |

**CREATE TABLE EXTERNAL\_INTERVIEW**

|  |  |  |
| --- | --- | --- |
| **(External\_ID** | **CHAR (9)** | **NOT NULL,** |
| **Interviewer\_ID** | **CHAR (9)** | **NOT NULL,** |
| **Job\_num** | **INT** | **NOT NULL,** |
| **Round** | **INT** | **NOT NULL,** |
| **Grade** | **INT** | **NOT NULL,** |
| **Time** | **DATE** | **NOT NULL,** |
| **PRIMARY KEY (External\_ID, interviewer\_ID, job\_num, round),** | | |
| **FOREIGN KEY (External \_ID) REFERENCES POTENTIAL\_EMPLOYEE (Pssn)** | | |
| **ON DELETE CASCADE                        ON UPDATE CASCADE,** | | |
| **FOREIGN KEY (Interviewer\_ID) REFERENCES EMPLOYEE (Ssn)** | | |
| **ON DELETE CASCADE                        ON UPDATE CASCADE,** | | |
| **FOREIGN KEY (Job\_num) REFERENCES JOB (Job\_ID)** | | |
| **ON DELETE CASCADE        ON UPDATE CASCADE);** | | |

**CREATE TABLE INTERNAL\_INTERVIEW**

|  |  |  |
| --- | --- | --- |
| **(Internal\_ID** | **CHAR (9)** | **NOT NULL,** |
| **Interviewer\_ID** | **CHAR (9)** | **NOT NULL,** |
| **Job\_num** | **INT** | **NOT NULL,** |
| **Round** | **INT** | **NOT NULL,** |
| **Grade** | **INT** | **NOT NULL,** |
| **Time** | **DATE** | **NOT NULL,** |
| **PRIMARY KEY (Internal\_ID, interviewer\_ID, job\_num, round),** | | |
| **FOREIGN KEY (Internal \_ID) REFERENCES EMPLOYEE (Ssn)** | | |
| **ON DELETE CASCADE                        ON UPDATE CASCADE,** | | |
| **FOREIGN KEY (Interviewer\_ID) REFERENCES EMPLOYEE (Ssn)** | | |
| **ON DELETE CASCADE                        ON UPDATE CASCADE,** | | |
| **FOREIGN KEY (Job\_num) REFERENCES JOB (Job\_ID)** | | |
| **ON DELETE CASCADE        ON UPDATE CASCADE);** | | |

**CREATE TABLE MARKETING\_SITE**

|  |  |  |
| --- | --- | --- |
| **( Site\_ID** | **INT** | **NOT NULL,** |
| **Site\_name** | **VARCHAR (15)** | **NOT NULL,** |
| **Site\_location** | **VARCHAR (15)** | **NOT NULL,** |
| **PRIMARY KEY (Site\_ID));** | | |

**CREATE TABLE SALE\_RECCORD**

|  |  |  |
| --- | --- | --- |
| **( Salesmen\_ID** | **CHAR (9)** | **NOT NULL,** |
| **Cust\_ID** | **CHAR (9)** | **NOT NULL,** |
| **Product\_number** | **INT** | **NOT NULL,** |
| **Site\_number** | **INT** | **NOT NULL,** |
| **Sales\_time** | **VARCHAR (15)** | **NOT NULL,** |
| **PRIMARY KEY (Salesmen\_ID, Cust\_ID, Product\_number, Site\_number),** | | |
| **FOREIGN KEY (Salesmen\_ID) REFERENCES EMPLOYEE (Ssn)** | | |
| **ON DELETE CASCADE                        ON UPDATE CASCADE,** | | |
| **FOREIGN KEY (Cust\_ID) REFERENCES CUSTOMER (Cssn)** | | |
| **ON DELETE CASCADE                        ON UPDATE CASCADE,** | | |
| **FOREIGN KEY (Product\_number) REFERENCES PRODUCT (Product\_ID)** | | |
| **ON DELETE CASCADE                        ON UPDATE CASCADE,** | | |
| **FOREIGN KEY (Site\_number) REFERENCES MARKETING\_SITE (Site\_ID)** | | |
| **ON DELETE CASCADE           ON UPDATE CASCADE);** | | |

**CREATE TABLE PRODUCT**

|  |  |  |
| --- | --- | --- |
| **( Product\_ID** | **INT** | **NOT NULL,** |
| **Product\_type** | **VARCHAR (30),** |  |
| **Size** | **CHAR (10),** |  |
| **List\_price** | **DECIMAL (10,2),** |  |
| **Weight** | **DECIMAL (8,2),** |  |
| **Style** | **VARCHAR (15),** |  |
| **PRIMARY KEY (Product\_ID));** | | |

**CREATE TABLE PART**

|  |  |  |
| --- | --- | --- |
| **( Part\_ID** | **INT** | **NOT NULL,** |
| **Part\_name** | **VARCHAR (15),** |  |
| **Part\_type** | **VARCHAR (15),** |  |
| **Part\_weight** | **DECIMAL (8,2),** |  |
| **PRIMARY KEY (Part\_ID));** | | |

**CREATE TABLE VENDOR**

|  |  |  |
| --- | --- | --- |
| **( Vendor\_ID** | **INT** | **NOT NULL,** |
| **Vname** | **VARCHAR (15),** |  |
| **Vaddress** | **VARCHAR (30),** |  |
| **Account\_number** | **VARCHAR (12),** |  |
| **Credit\_rating** | **CHAR (5),** |  |
| **Web\_URL** | **VARCHAR (50),** |  |
| **PRIMARY KEY (Vendor\_ID));** | | |

**CREATE TABLE SUPPLY**

|  |  |  |
| --- | --- | --- |
| **(Vnumber** | **INT** | **NOT NULL,** |
| **Pnum** | **INT** | **NOT NULL,** |
| **Part\_number** | **INT** | **NOT NULL,** |
| **Quantity** | **INT,** |  |
| **Part\_price** | **DECIMAL (10,2),** |  |
| **PRIMARY KEY (Vnumber, Product\_num, Part\_number),** | | |
| **FOREIGN KEY (Vnumber) REFERENCES VENDOR (Vendor\_ID)** | | |
| **ON DELETE CASCADE                        ON UPDATE CASCADE,** | | |
| **FOREIGN KEY (Product\_num) REFERENCES PRODUCT (Product\_ID)** | | |
| **ON DELETE CASCADE                        ON UPDATE CASCADE,** | | |
| **FOREIGN KEY (Part\_number) REFERENCES PART (Part\_ID)** | | |
| **ON DELETE SET CASCADE        ON UPDATE CASCADE);** | | |

**VIEW:**

**1.**

**CREATE VIEW `View1` AS**

**SELECT Essn, AVG(Amount)**

**FROM MONTHLY\_SALARY**

**GROUP BY Essn**

**2.**

**CREATE VIEW `view2` AS**

**(**

**SELECT Internal\_ID, Job\_num, COUNT(\*) AS TOTAL\_PASSED**

**FROM INTERNAL\_INTERVIEW**

**WHERE Grade > 60**

**GROUP BY Internal\_ID, Job\_num**

**UNION**

**SELECT External\_ID, Job\_num, COUNT(\*) AS TOTAL\_PASSED**

**FROM EXTERNAL\_INTERVIEW**

**WHERE Grade > 60**

**GROUP BY External\_ID, Job\_num)**

**3.**

**CREATE VIEW `view3` AS**

**SELECT Product\_number, Product\_type, COUNT(\*) AS NUMBER\_SOLD**

**FROM SALE\_RECORD, PRODUCT**

**WHERE Product\_number = Product\_ID**

**GROUP BY Product\_number, Product\_type**

**4.**

**CREATE VIEW `view4` AS**

**SELECT T.Product\_number, P.Product\_type, (MIN(S.Part\_Price)\*T.Quantity) AS PART\_COST**

**FROM PRODUCT P, PRODUCT\_PART T, SUPPLY S**

**WHERE T.Product\_number = P.Product\_ID AND T.Part\_number = S.Part\_number**

**GROUP BY T.Product\_number, P.Product\_type, T.Quantity**

**SQL:**

**1.**

**SELECT DISTINCT E.Ssn, E.Fname, E.Lname**

**FROM EMPLOYEE E, EXTERNAL\_INTERVIEW X**

**WHERE E.Ssn = X.Interviewer\_ID AND X.External\_ID IN**

**(**

**SELECT P.Pssn**

**FROM POTENTIAL\_EMPLOYEE P**

**WHERE P.Fname = "Hellen" AND P.Lname ="Code")**

**AND X.Job\_num = "11111"**

**UNION**

**SELECT DISTINCT E.Ssn, E.Fname, E.Lname**

**FROM EMPLOYEE E, INTERNAL\_INTERVIEW I**

**WHERE E.Ssn = I.Interviewer\_ID AND I.Internal\_ID IN**

**(**

**SELECT E.Ssn**

**FROM EMPLOYEE E**

**WHERE E.Fname = "Hellen" AND E.Lname ="Code")**

**AND I.Job\_num = "11111";**

**2.**

**SELECT J.Job\_ID**

**FROM JOB J, DEPARTMENT D**

**WHERE J.Dept\_ID = D.Dnumber AND D.Dname = "Marketing";**

**3.**

**SELECT E.Ssn, E.Fname, E.Lname**

**FROM EMPLOYEE E**

**WHERE E.Ssn NOT IN (SELECT E.Super\_ssn**

**FROM EMPLOYEE E**

**WHERE E.Super\_ssn IS NOT NULL);**

**4.**

**SELECT DISTINCT S.Site\_ID, S.Site\_location**

**FROM MARKETING\_SITE S, SALE\_RECORD R**

**WHERE S.Site\_ID = R.Site\_number**

**AND R.Sale\_time NOT BETWEEN "2011-03-01" AND "2011-03-31";**

**5.**

**SELECT J.Job\_ID**

**FROM JOB J**

**LEFT JOIN**

**(SELECT I.Internal\_ID AS ID, I.Job\_num, AVG(I.GRADE) AS AVERAGE\_GRADE, COUNT(\*) as TOTAL\_INTERVIEW**

**FROM INTERNAL\_INTERVIEW I, JOB J**

**WHERE J.Job\_ID = I.Job\_num AND I.GRADE > 60 AND I.Time BETWEEN J.Job\_date AND DATE\_ADD(J.Job\_date, INTERVAL 1 MONTH)**

**GROUP BY I.Internal\_ID, I.Job\_num**

**HAVING AVERAGE\_GRADE > 70 AND TOTAL\_INTERVIEW >= 5**

**UNION**

**SELECT E.External\_ID AS ID, E.Job\_num, AVG(E.GRADE) AS AVERAGE\_GRADE, COUNT(\*) as TOTAL\_INTERVIEW**

**FROM EXTERNAL\_INTERVIEW E, JOB J**

**WHERE J.Job\_ID = E.Job\_num AND E.GRADE > 60 AND E.Time BETWEEN J.Job\_date AND DATE\_ADD(J.Job\_date, INTERVAL 1 MONTH)**

**GROUP BY E.External\_ID, E.Job\_num**

**HAVING AVERAGE\_GRADE > 70 AND TOTAL\_INTERVIEW >= 5) A**

**ON J.Job\_ID = A.Job\_num**

**WHERE A.Job\_num IS NULL**

**6.**

**SELECT DISTINCT E.Ssn, E.Fname, E.Lname**

**FROM EMPLOYEE E, SALE\_RECORD R, PRODUCT P**

**WHERE E.Ssn = R.Salesmen\_ID AND R.Product\_number = P.Product\_ID AND P.List\_price > 200;**

**7.**

**SELECT DISTINCT D.Dnumber, D.Dname**

**FROM DEPARTMENT D, JOB J**

**WHERE J.Dept\_ID = D.Dnumber AND J.Job\_date NOT BETWEEN "2011-01-01" AND "2011-02-01";**

**8.**

**SELECT DISTINCT E.Ssn, E.Lname, E.Fname, E.Dno**

**FROM EMPLOYEE E, INTERNAL\_INTERVIEW I**

**WHERE I.Internal\_ID = E.Ssn AND I.Job\_num = "12345";**

**9.**

**SELECT P.Product\_type, COUNT(\*) AS TOTAL\_SALE**

**FROM PRODUCT P, SALE\_RECORD S**

**WHERE P.Product\_ID = S.Product\_number**

**GROUP BY P.Product\_type**

**ORDER BY TOTAL\_SALE DESC LIMIT 1**

**10.**

**SELECT P.Product\_number, T.Product\_type,**

**MIN(P.Quantity\*S.Part\_price) AS Part\_Cost,**

**T.List\_price - MIN(P.Quantity\*S.Part\_price) AS Profit**

**FROM PRODUCT\_PART P, SUPPLY S, PRODUCT T**

**WHERE P.Part\_number = S.Part\_number AND P.Product\_number = T.Product\_ID**

**GROUP BY P.Product\_number**

**ORDER BY Profit DESC LIMIT 1**

**11.**

**SELECT E.Fname, E.Lname, E.Ssn, COUNT(\*) AS TOTAL\_DEPARTMENT**

**FROM EMPLOYEE E, WORK\_FOR W, DEPARTMENT D**

**WHERE E.Ssn = W.Emp\_ID AND D.Dnumber = W.Dnum**

**GROUP BY E.Ssn**

**HAVING TOTAL\_DEPARTMENT = (SELECT COUNT(\*) FROM DEPARTMENT)**

**12.**

**SELECT E.Fname, E.Lname, E.Email**

**FROM EMPLOYEE E,**

**(SELECT I.Internal\_ID AS ID, I.Job\_num, AVG(I.GRADE) AS AVERAGE\_GRADE, COUNT(\*) as TOTAL\_INTERVIEW**

**FROM INTERNAL\_INTERVIEW I, JOB J**

**WHERE J.Job\_ID = I.Job\_num AND I.GRADE > 60 AND I.Time BETWEEN J.Job\_date AND DATE\_ADD(J.Job\_date, INTERVAL 1 MONTH)**

**GROUP BY I.Internal\_ID, I.Job\_num**

**HAVING AVERAGE\_GRADE > 70 AND TOTAL\_INTERVIEW >= 5**

**UNION**

**SELECT E.External\_ID AS ID, E.Job\_num, AVG(E.GRADE) AS AVERAGE\_GRADE, COUNT(\*) as TOTAL\_INTERVIEW**

**FROM EXTERNAL\_INTERVIEW E, JOB J**

**WHERE J.Job\_ID = E.Job\_num AND E.GRADE > 60 AND E.Time BETWEEN J.Job\_date AND DATE\_ADD(J.Job\_date, INTERVAL 1 MONTH)**

**GROUP BY E.External\_ID, E.Job\_num**

**HAVING AVERAGE\_GRADE > 70 AND TOTAL\_INTERVIEW >= 5) A**

**WHERE E.Ssn = A.ID**

**UNION**

**SELECT P.Fname, P.Lname, P.Email**

**FROM POTENTIAL\_EMPLOYEE P,**

**(SELECT I.Internal\_ID AS ID, I.Job\_num, AVG(I.GRADE) AS AVERAGE\_GRADE, COUNT(\*) as TOTAL\_INTERVIEW**

**FROM INTERNAL\_INTERVIEW I, JOB J**

**WHERE J.Job\_ID = I.Job\_num AND I.GRADE > 60 AND I.Time BETWEEN J.Job\_date AND DATE\_ADD(J.Job\_date, INTERVAL 1 MONTH)**

**GROUP BY I.Internal\_ID, I.Job\_num**

**HAVING AVERAGE\_GRADE > 70 AND TOTAL\_INTERVIEW >= 5**

**UNION**

**SELECT E.External\_ID AS ID, E.Job\_num, AVG(E.GRADE) AS AVERAGE\_GRADE, COUNT(\*) as TOTAL\_INTERVIEW**

**FROM EXTERNAL\_INTERVIEW E, JOB J**

**WHERE J.Job\_ID = E.Job\_num AND E.GRADE > 60 AND E.Time BETWEEN J.Job\_date AND DATE\_ADD(J.Job\_date, INTERVAL 1 MONTH)**

**GROUP BY E.External\_ID, E.Job\_num**

**HAVING AVERAGE\_GRADE > 70 AND TOTAL\_INTERVIEW >= 5) A**

**WHERE P.Pssn = A.ID**

**13.**

**SELECT E.Fname, E.Lname, E.Phone, E.Email**

**FROM EMPLOYEE E,**

**(SELECT A.ID, COUNT(A.ID) AS TOTAL\_PASSED**

**FROM**

**(SELECT I.Internal\_ID AS ID, I.Job\_num, AVG(I.GRADE) AS AVERAGE\_GRADE, COUNT(\*) as TOTAL\_INTERVIEW**

**FROM INTERNAL\_INTERVIEW I, JOB J**

**WHERE J.Job\_ID = I.Job\_num AND I.GRADE > 60 AND I.Time BETWEEN J.Job\_date AND DATE\_ADD(J.Job\_date, INTERVAL 1 MONTH)**

**GROUP BY I.Internal\_ID, I.Job\_num**

**HAVING AVERAGE\_GRADE > 70 AND TOTAL\_INTERVIEW >= 5**

**UNION**

**SELECT E.External\_ID AS ID, E.Job\_num, AVG(E.GRADE) AS AVERAGE\_GRADE, COUNT(\*) as TOTAL\_INTERVIEW**

**FROM EXTERNAL\_INTERVIEW E, JOB J**

**WHERE J.Job\_ID = E.Job\_num AND E.GRADE > 60 AND E.Time BETWEEN J.Job\_date AND DATE\_ADD(J.Job\_date, INTERVAL 1 MONTH)**

**GROUP BY E.External\_ID, E.Job\_num**

**HAVING AVERAGE\_GRADE > 70 AND TOTAL\_INTERVIEW >= 5) A**

**GROUP BY A.ID) A,**

**(SELECT A.ID, COUNT(A.Job\_num) AS TOTAL\_APPLIED**

**FROM(**

**SELECT Internal\_ID AS ID, Job\_num**

**FROM INTERNAL\_INTERVIEW**

**GROUP BY Internal\_ID, Job\_num**

**UNION**

**SELECT External\_ID AS ID, Job\_num**

**FROM EXTERNAL\_INTERVIEW**

**GROUP BY External\_ID, Job\_num**

**) A**

**GROUP BY A.ID) B**

**WHERE A.TOTAL\_PASSED = B.TOTAL\_APPLIED AND A.ID = B.ID AND E.Ssn = A.ID**

**UNION**

**SELECT P.Fname, P.Lname, P.Phone, P.Email**

**FROM POTENTIAL\_EMPLOYEE P,**

**(SELECT A.ID, COUNT(A.ID) AS TOTAL\_PASSED**

**FROM**

**(SELECT I.Internal\_ID AS ID, I.Job\_num, AVG(I.GRADE) AS AVERAGE\_GRADE, COUNT(\*) as TOTAL\_INTERVIEW**

**FROM INTERNAL\_INTERVIEW I, JOB J**

**WHERE J.Job\_ID = I.Job\_num AND I.GRADE > 60 AND I.Time BETWEEN J.Job\_date AND DATE\_ADD(J.Job\_date, INTERVAL 1 MONTH)**

**GROUP BY I.Internal\_ID, I.Job\_num**

**HAVING AVERAGE\_GRADE > 70 AND TOTAL\_INTERVIEW >= 5**

**UNION**

**SELECT E.External\_ID AS ID, E.Job\_num, AVG(E.GRADE) AS AVERAGE\_GRADE, COUNT(\*) as TOTAL\_INTERVIEW**

**FROM EXTERNAL\_INTERVIEW E, JOB J**

**WHERE J.Job\_ID = E.Job\_num AND E.GRADE > 60 AND E.Time BETWEEN J.Job\_date AND DATE\_ADD(J.Job\_date, INTERVAL 1 MONTH)**

**GROUP BY E.External\_ID, E.Job\_num**

**HAVING AVERAGE\_GRADE > 70 AND TOTAL\_INTERVIEW >= 5) A**

**GROUP BY A.ID) A,**

**(SELECT A.ID, COUNT(A.Job\_num) AS TOTAL\_APPLIED**

**FROM(**

**SELECT Internal\_ID AS ID, Job\_num**

**FROM INTERNAL\_INTERVIEW**

**GROUP BY Internal\_ID, Job\_num**

**UNION**

**SELECT External\_ID AS ID, Job\_num**

**FROM EXTERNAL\_INTERVIEW**

**GROUP BY External\_ID, Job\_num**

**) A**

**GROUP BY A.ID) B**

**WHERE A.TOTAL\_PASSED = B.TOTAL\_APPLIED AND A.ID = B.ID AND P.Pssn = A.ID**

**14.**

**SELECT E.Fname, E.Lname, E.Ssn, AVG(M.Amount) AS Average\_Monthly\_Salary**

**FROM EMPLOYEE E, MONTHLY\_SALARY M**

**WHERE E.Ssn = M.Essn**

**GROUP BY E.Ssn**

**ORDER BY Average\_Monthly\_Salary DESC LIMIT 1**

**15.**

**SELECT V.Vendor\_ID, V.Vname, P.Part\_name, MIN(S.Part\_price) AS MIN\_PRICE**

**FROM VENDOR V, SUPPLY S, PART P**

**WHERE V.Vendor\_ID = S.Vnumber AND S.Part\_number = P.Part\_ID AND P.Part\_name = "CUP" AND P.Part\_weight < 4.00**

**GROUP BY V.Vendor\_ID**

**ORDER BY MIN\_PRICE ASC LIMIT 1**

**g) Dependency diagram.**

Diagram

Description automatically generated