**ACCT 6321 Database Applications for Business Analytics in Accounting**

**Fall 2022**

**Instructor: Dr. James Scott**

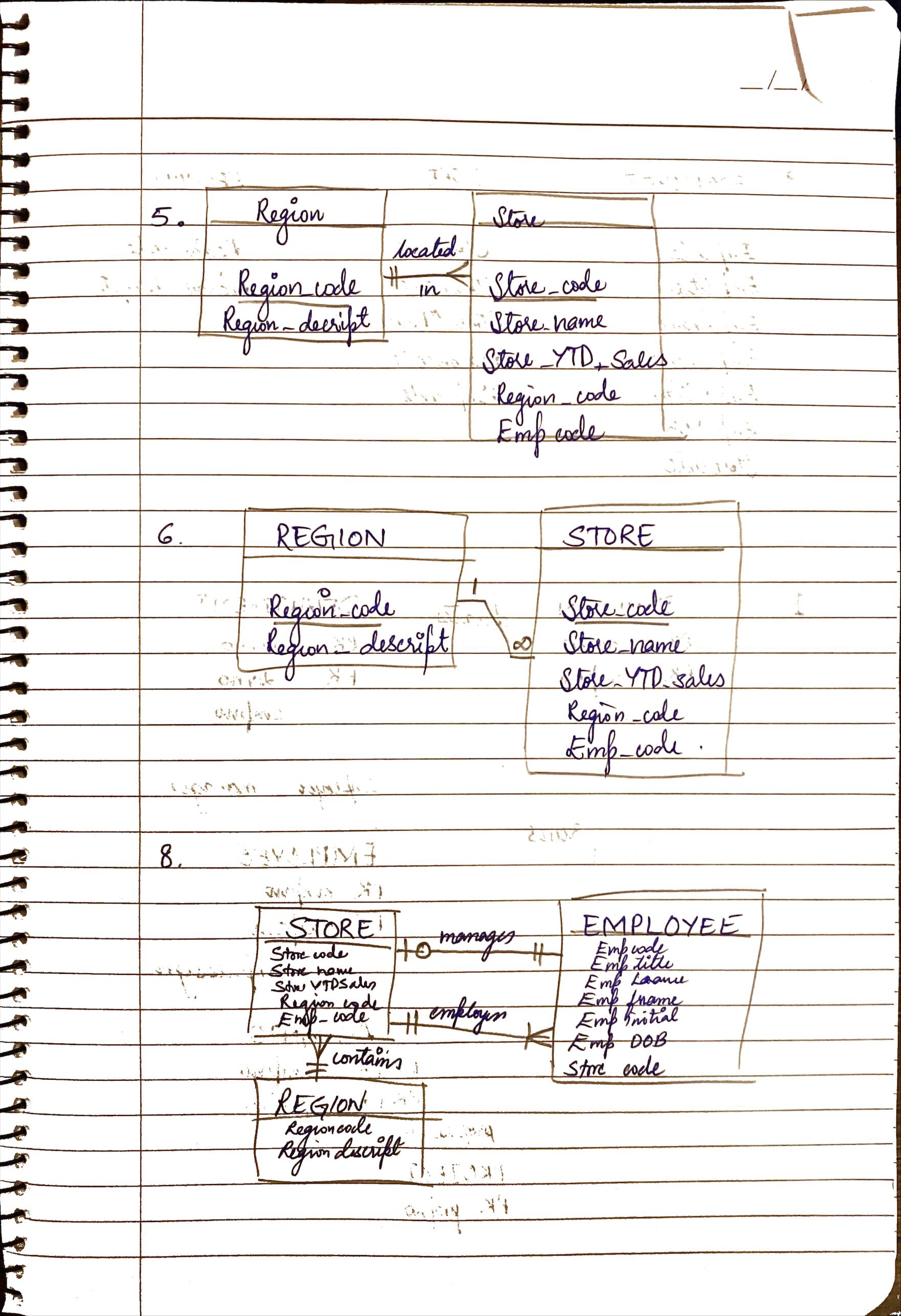
**Assignment #2 – 100 Points**

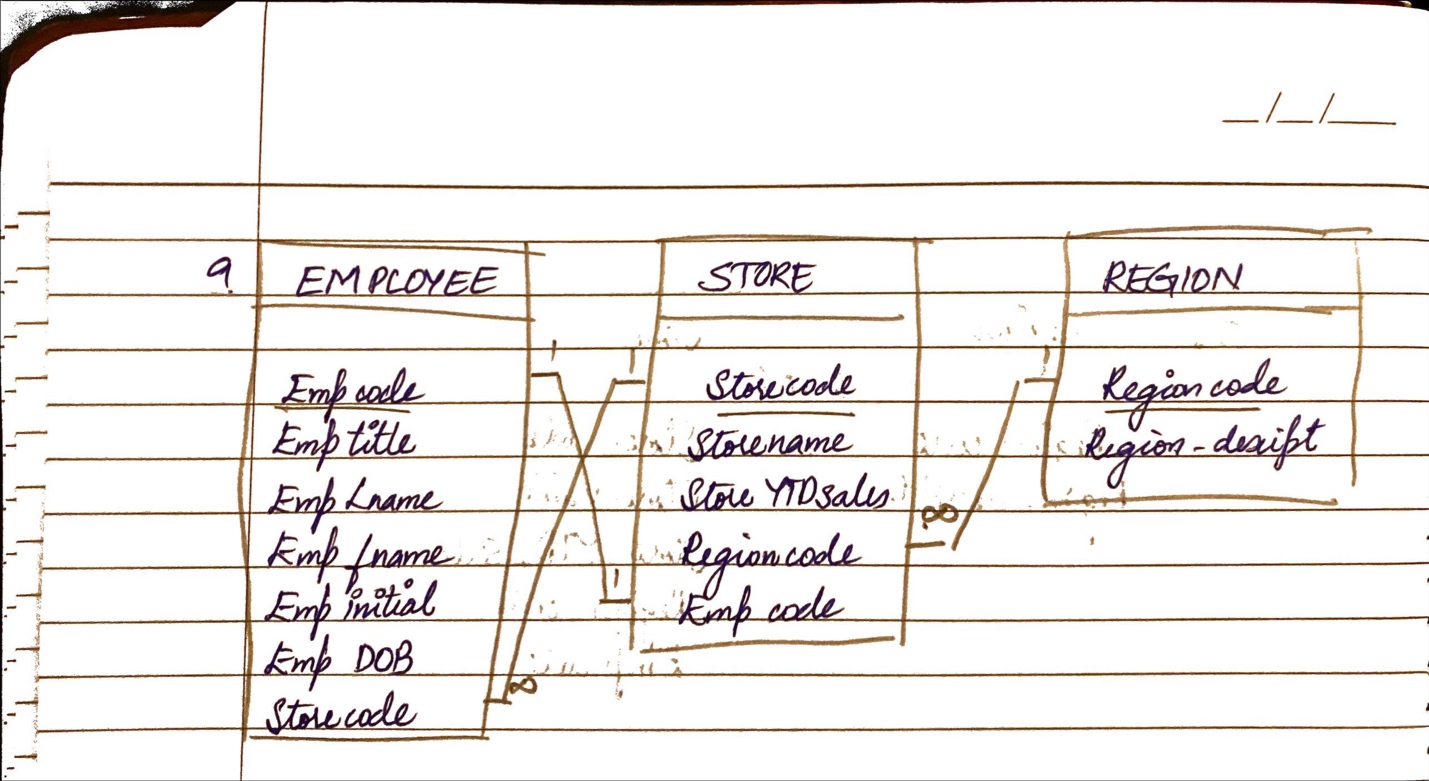
**General Instructions**

* Students may study together for the assignment and review each other’s completed work
* Students must each complete the assignment by their own hand
* Please use the provided word document template
* Please save the completed word document into PDF format before uploading
* Please submit the PDF file electronically through eLearning before the due date and time
* Do not worry about variations among database vendors – you may write SQL to any vendor’s dialect
* Do not include output – only the SQL
* **Use table aliases for all tables in all queries (unless otherwise specified)**
* Column aliases are required for all derived columns including aggregate columns (unless otherwise specified)
* Do not use column aliases unless required as stated previously
* If a problem does not ask for a specific sort order, use your best judgement to add a sort order

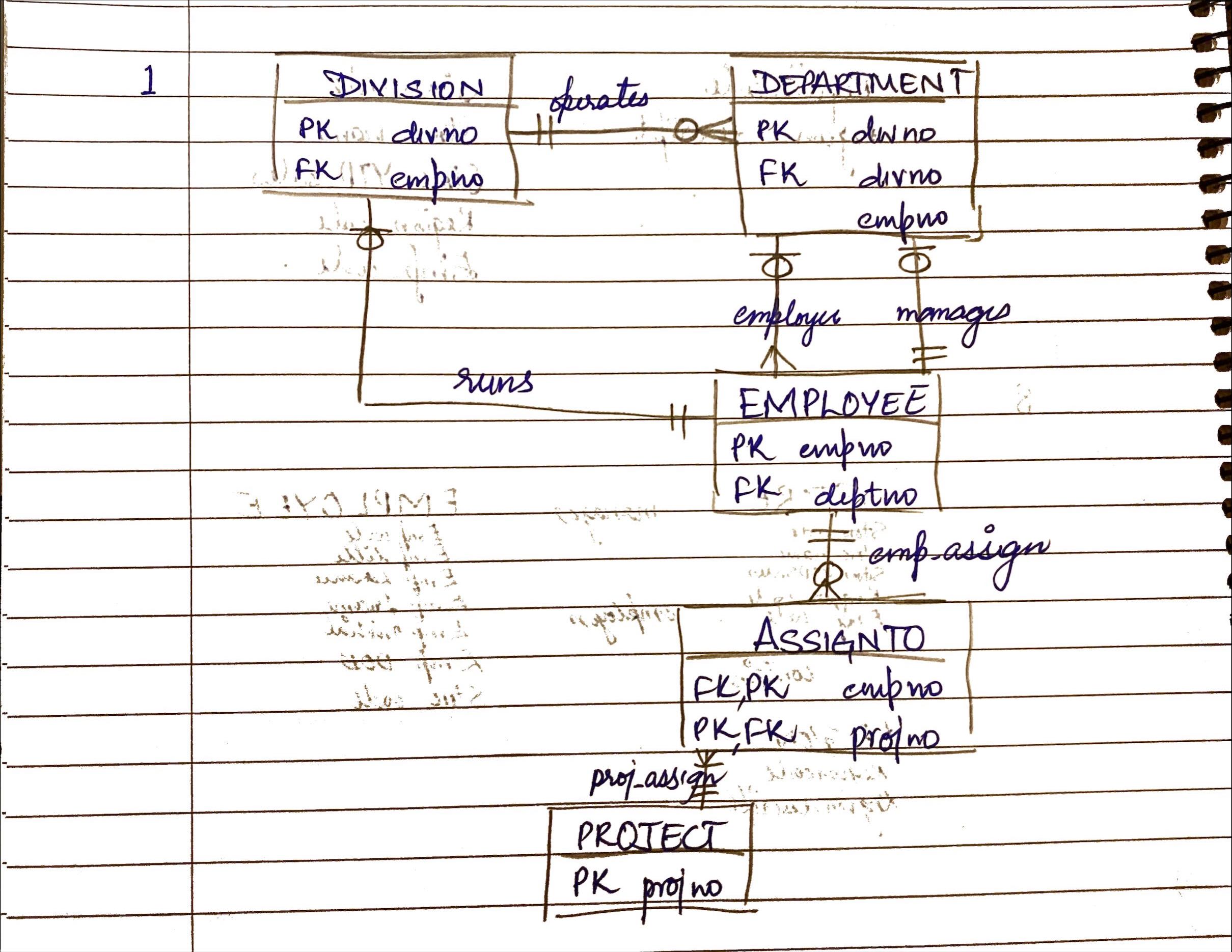
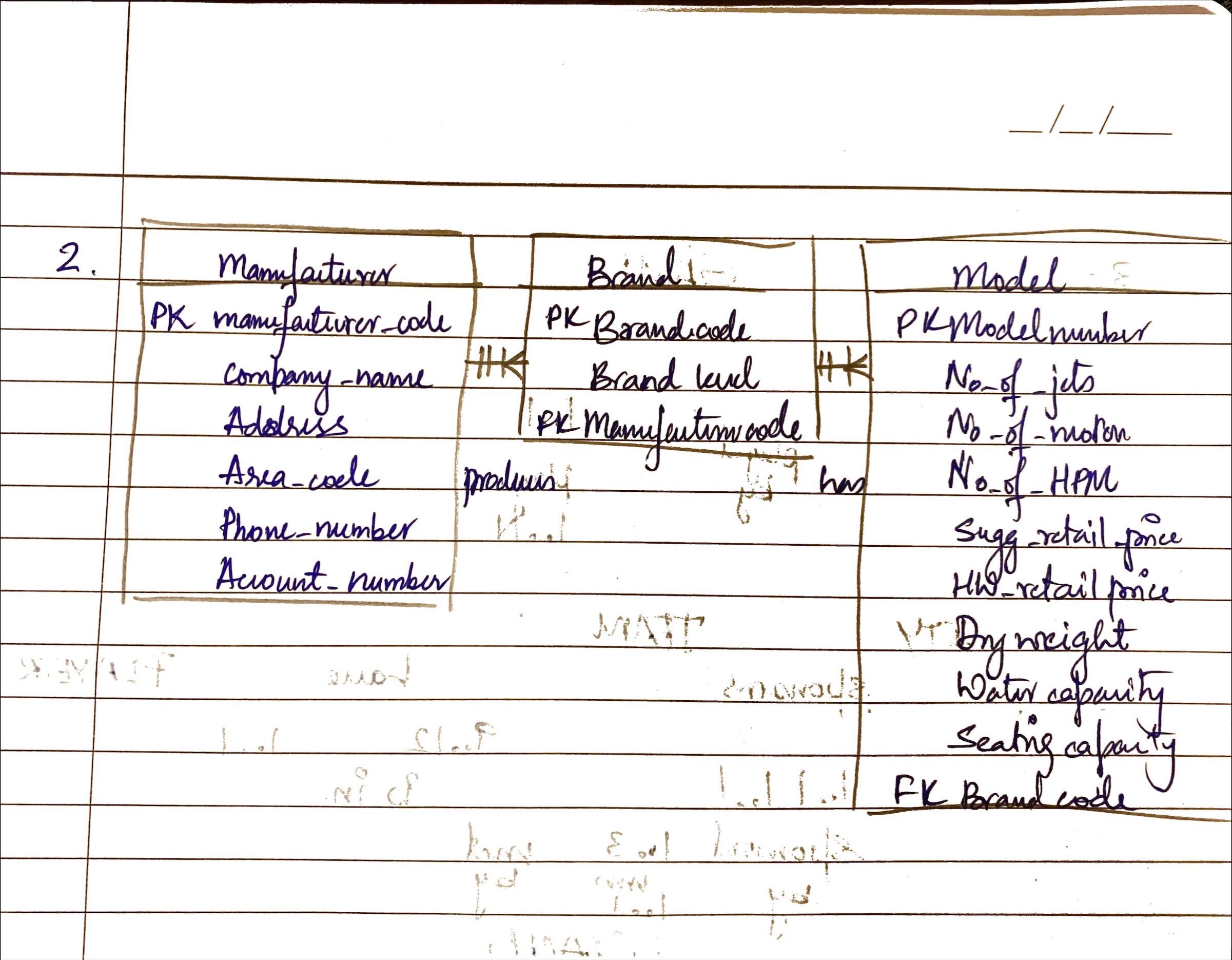
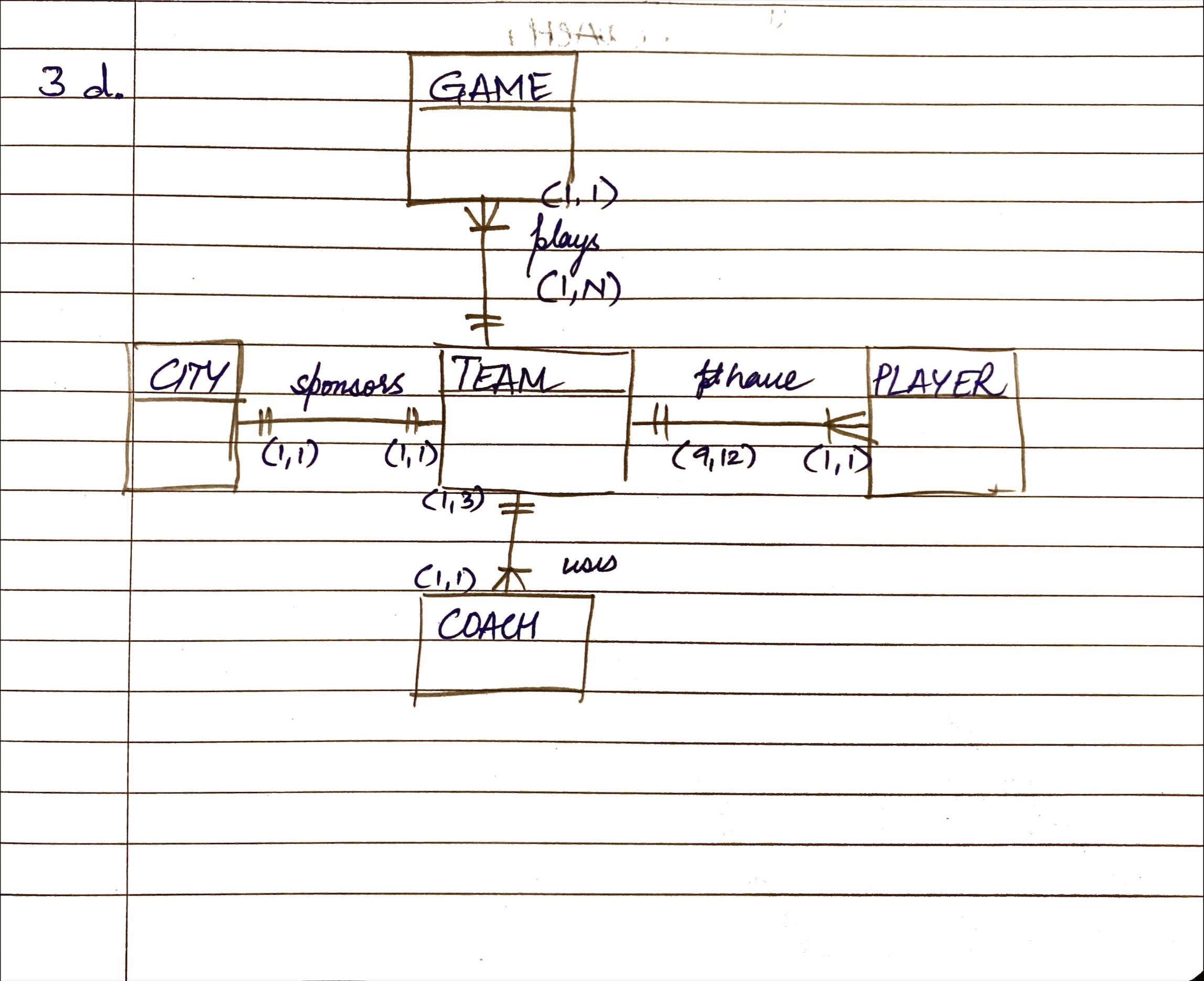
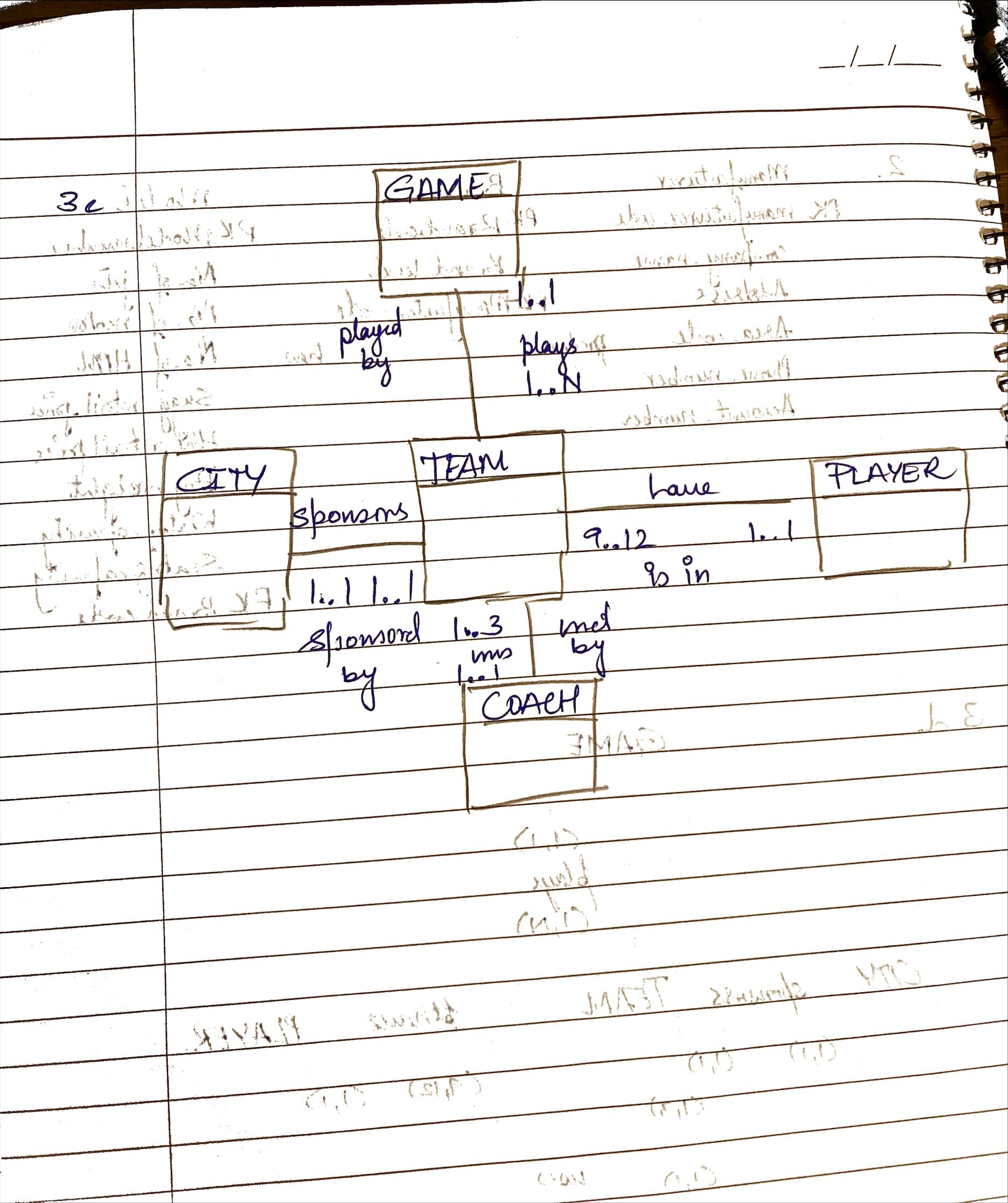
**Chapter 3 Problems – The Relational Database Model**

1. The tables have the following parameters:
   1. EMPLOYEE:
      1. PRIMARY KEY: EMP\_CODE
      2. FORGIEN KEY: STORE\_CODE
   2. STORE:
      1. PRIMARY KEY: STORE\_CODE
      2. FORGIEN KEY: REGION\_CODE
   3. REGION:
      1. PRIMARY KEY: REGION\_CODE
      2. FORGIEN KEY : None
2. All the tables have table integrity as they all have non null and unique primary key.
3. The following tables:
   1. EMPLOYEE: Yes, it exhibits referential integrity as it has a forgien key ‘STORE\_CODE’.
   2. STORE: Yes, it exhibits referential integrity as it has a forgien key ‘REGION\_CODE & EMP\_CODE’.
   3. REGION: NA
4. The relationship between STORE and REGION is 1:M as there are multiple stores in a specific region but only one region for a specific store.
5. Please find diagram below:
6. Please find diagram below:
7. The relationship between EMPLOYEE and STORE is 1:M as there are multiple emplyees at a given store, but an employee can only work at one store.
8. Please find diagram below:





**Chapter 4 Problems – Entity Relationship (ER) Modeling**

1. 
2. 
3. Answers are as follows:
   1. These are the following relationships:
      1. CITY and TEAM have a 1:1 relationship as each city can only one team.
      2. COACH and TEAM have a 1:M relationship as each team can have 3 coacches.
      3. TEAM and PLAYER have a 1:M relationship as each team has 9-12 players.
   2. TEAM has a existence depedent on CITY
   3. The cardinality are as follows:
      1. TEAM and PLAYER : each team can have minimum of nine players and maximum of 12 players having (9, 12) cardinality
      2. TEAM and PLAYER: Each player can be only in one team having cardinality of (1,1).
   4. 
   5. 

**Using the following University DDL code for MySQL and using MySQL with DBeaver, please create the database, tables, and insert the accompanying data to answer the following eight (8-15) practical SQL questions that follo**w**. You may create the database using the SQL code or using DBeaver.**

-- Script: University\_MySQL DDL

-- MySQL/Maria Good

-- DROP TABLE IF EXISTS Faculty

**CREATE** **TABLE** Faculty (

FacNo **char**(11) **NOT** **NULL**,

FacFirstName **varchar**(30) **NOT** **NULL**,

FacLastName **varchar**(30) **NOT** **NULL**,

FacCity **varchar**(30) **NOT** **NULL**,

FacState **char**(2) **NOT** **NULL**,

FacDept **char**(6) **DEFAULT** **NULL**,

FacRank **char**(4) **DEFAULT** **NULL**,

FacSalary **decimal**(10,2) **DEFAULT** **NULL**,

FacSupervisor **char**(11) **DEFAULT** **NULL**,

FacHireDate **datetime** **DEFAULT** **NULL**,

FacZipCode **char**(10) **NOT** **NULL**,

**PRIMARY** **KEY** (FacNo),

**KEY** FacSupervisor\_idx (FacSupervisor)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;

-- DROP TABLE IF EXISTS Course;

**CREATE** **TABLE** Course (

CourseNo **CHAR**(6) **NOT** **NULL**,

CrsDesc **VARCHAR**(50) **NOT** **NULL**,

CrsUnits **INTEGER** **NULL**,

**CONSTRAINT** CoursePK **PRIMARY** **KEY** (CourseNo)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;

-- DROP TABLE IF EXISTS Offering;

**CREATE** **TABLE** Offering (

OfferNo **CHAR**(6) **NOT** **NULL**,

CourseNo **CHAR**(6) **NOT** **NULL**,

OffTerm **CHAR**(6) **NOT** **NULL**,

OffYear **INT** **NOT** **null**,

OffLocation **VARCHAR**(30) **NULL**,

OffTime **VARCHAR**(10) **null**,

FacNo **CHAR**(11) **NULL**,

OffDays **CHAR**(4) **NULL**,

**CONSTRAINT** OfferingPK **PRIMARY** **KEY** (OfferNo),

**CONSTRAINT** CourseFk **FOREIGN** **KEY** (CourseNo) **REFERENCES** Course(CourseNo),

**CONSTRAINT** FacultyFK **FOREIGN** **KEY** (FacNo) **REFERENCES** faculty(FacNo)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;

-- DROP TABLE IF EXISTS Student;

**CREATE** **TABLE** Student (

StdNo **CHAR**(11) **NOT** **NULL**,

StdFirstName **VARCHAR**(30) **NOT** **NULL**,

StdLastName **VARCHAR**(30) **NOT** **NULL**,

StdCity **VARCHAR**(30) **NOT** **NULL**,

StdState **CHAR**(2) **NOT** **NULL**,

StdZip **CHAR**(10) **NOT** **NULL**,

StdMajor **CHAR**(6) **NULL**,

StdClass **CHAR**(2) **NULL**,

StdGPA **DECIMAL**(3,2) **NULL**,

**CONSTRAINT** StudentPk **PRIMARY** **KEY** (StdNo)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;

-- DROP TABLE IF EXISTS Enrollment;

**CREATE** **TABLE** Enrollment(

OfferNo **CHAR**(6) **NOT** **NULL**,

StdNo **CHAR**(11) **NOT** **NULL**,

EnrGrade **DECIMAL**(3,2) **NULL**,

**CONSTRAINT** EnrollmentPK **PRIMARY** **KEY** (OfferNo, StdNo),

**CONSTRAINT** OfferingFK **FOREIGN** **KEY** (OfferNo) **REFERENCES** Offering(OfferNo),

**CONSTRAINT** StudentFK **FOREIGN** **KEY** (StdNo) **REFERENCES** Student(StdNo)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;

**INSERT** **INTO** Faculty **VALUES**

('543210987','VICTORIA','EMMANUEL','BOTHELL','WA','MS','PROF',120000.00,**NULL**,STR\_TO\_DATE('4/15/1998','%m/%d/%Y'),'98011-2242');

**INSERT** **INTO** Faculty **VALUES**

('654321098','LEONARD','FIBON','SEATTLE','WA','MS','ASSC',70000.00,'543210987',STR\_TO\_DATE('5/1/1996','%m/%d/%Y'),'98121-0094');

**INSERT** **INTO** Faculty **VALUES**

('098765432','LEONARD','VINCE','SEATTLE','WA','MS','ASST',35000.00,'654321098',STR\_TO\_DATE('4/10/1997','%m/%d/%Y'),'98111-9921');

**INSERT** **INTO** Faculty **VALUES**

('765432109','NICKI','MACON','BELLEVUE','WA','FIN','PROF',65000.00,**NULL**,STR\_TO\_DATE('4/11/1999','%m/%d/%Y'),'98015-9945');

**INSERT** **INTO** Faculty **VALUES**

('876543210','CRISTOPHER','COLAN','SEATTLE','WA','MS','ASST',40000.00,'654321098',STR\_TO\_DATE('3/1/2001','%m/%d/%Y'),'98114-1332');

**INSERT** **INTO** Faculty **VALUES** ('987654321','JULIA','MILLS','SEATTLE','WA','FIN','ASSC',75000.00,'765432109',STR\_TO\_DATE('3/15/2002','%m/%d/%Y'),'98114-9954');

**INSERT** **INTO** Course **VALUES** ('FIN300','FUNDAMENTALS OF FINANCE',4);

**INSERT** **INTO** Course **VALUES** ('FIN450','PRINCIPLES OF INVESTMENTS',4);

**INSERT** **INTO** Course **VALUES** ('FIN480','CORPORATE FINANCE',4);

**INSERT** **INTO** Course **VALUES** ('IS320','FUNDAMENTALS OF BUSINESS PROGRAMMING',4);

**INSERT** **INTO** Course **VALUES** ('IS460','SYSTEMS ANALYSIS',4);

**INSERT** **INTO** Course **VALUES** ('IS470','BUSINESS DATA COMMUNICATIONS',4);

**INSERT** **INTO** Course **VALUES** ('IS480','FUNDAMENTALS OF DATABASE MANAGEMENT',4);

**INSERT** **INTO** Offering **VALUES** (1111,'IS320','SUMMER',2010,'BLM302','10:30:00',**NULL**,'MW');

**INSERT** **INTO** Offering **VALUES** (1234,'IS320','FALL',2009,'BLM302','10:30:00','098765432','MW');

**INSERT** **INTO** Offering **VALUES** (2222,'IS460','SUMMER',2009,'BLM412','13:30:00',**NULL**,'TTH');

**INSERT** **INTO** Offering **VALUES** (3333,'IS320','SPRING',2010,'BLM214','8:30:00','098765432','MW');

**INSERT** **INTO** Offering **VALUES** (4321,'IS320','FALL',2009,'BLM214','15:30:00','098765432','TTH');

**INSERT** **INTO** Offering **VALUES** (4444,'IS320','WINTER',2010,'BLM302','15:30:00','543210987','TTH');

**INSERT** **INTO** Offering **VALUES** (5555,'FIN300','WINTER',2010,'BLM207','8:30:00','765432109','MW');

**INSERT** **INTO** Offering **VALUES** (5678,'IS480','WINTER',2010,'BLM302','10:30:00','987654321','MW');

**INSERT** **INTO** Offering **VALUES** (5679,'IS480','SPRING',2010,'BLM412','15:30:00','876543210','TTH');

**INSERT** **INTO** Offering **VALUES** (6666,'FIN450','WINTER',2010,'BLM212','10:30:00','987654321','TTH');

**INSERT** **INTO** Offering **VALUES** (7777,'FIN480','SPRING',2010,'BLM305','13:30:00','765432109','MW');

**INSERT** **INTO** Offering **VALUES** (8888,'IS320','SUMMER',2010,'BLM405','13:30:00','654321098','MW');

**INSERT** **INTO** Offering **VALUES** (9876,'IS460','SPRING',2010,'BLM307','13:30:00','654321098','TTH');

**INSERT** **INTO** Student **VALUES** ('123456789','HOMER','WELLS','SEATTLE','WA','98121-1111','IS','FR',3.00);

**INSERT** **INTO** Student **VALUES** ('124567890','BOB','NORBERT','BOTHELL','WA','98011-2121','FIN','JR',2.70);

**INSERT** **INTO** Student **VALUES** ('234567890','CANDY','KENDALL','TACOMA','WA','99042-3321','ACCT','JR',3.50);

**INSERT** **INTO** Student **VALUES** ('345678901','WALLY','KENDALL','SEATTLE','WA','98123-1141','IS','SR',2.80);

**INSERT** **INTO** Student **VALUES** ('456789012','JOE','ESTRADA','SEATTLE','WA','98121-2333','FIN','SR',3.20);

**INSERT** **INTO** Student **VALUES** ('567890123','MARIAH','DODGE','SEATTLE','WA','98114-0021','IS','JR',3.60);

**INSERT** **INTO** Student **VALUES** ('678901234','TESS','DODGE','REDMOND','WA','98116-2344','ACCT','SO',3.30);

**INSERT** **INTO** Student **VALUES** ('789012345','ROBERTO','MORALES','SEATTLE','WA','98121-2212','FIN','JR',2.50);

**INSERT** **INTO** Student **VALUES** ('876543210','CRISTOPHER','COLAN','SEATTLE','WA','98114-1332','IS','SR',4.00);

**INSERT** **INTO** Student **VALUES** ('890123456','LUKE','BRAZZI','SEATTLE','WA','98116-0021','IS','SR',2.20);

**INSERT** **INTO** Student **VALUES** ('901234567','WILLIAM','PILGRIM','BOTHELL','WA','98113-1885','IS','SO',3.80);

**INSERT** **INTO** Enrollment **VALUES** (1234,'123456789',3.30);

**INSERT** **INTO** Enrollment **VALUES** (1234,'234567890',3.50);

**INSERT** **INTO** Enrollment **VALUES** (1234,'345678901',3.20);

**INSERT** **INTO** Enrollment **VALUES** (1234,'456789012',3.10);

**INSERT** **INTO** Enrollment **VALUES** (1234,'567890123',3.80);

**INSERT** **INTO** Enrollment **VALUES** (1234,'678901234',3.40);

**INSERT** **INTO** Enrollment **VALUES** (4321,'123456789',3.50);

**INSERT** **INTO** Enrollment **VALUES** (4321,'124567890',3.20);

**INSERT** **INTO** Enrollment **VALUES** (4321,'789012345',3.50);

**INSERT** **INTO** Enrollment **VALUES** (4321,'876543210',3.10);

**INSERT** **INTO** Enrollment **VALUES** (4321,'890123456',3.40);

**INSERT** **INTO** Enrollment **VALUES** (4321,'901234567',3.10);

**INSERT** **INTO** Enrollment **VALUES** (5555,'123456789',3.20);

**INSERT** **INTO** Enrollment **VALUES** (5555,'124567890',2.70);

**INSERT** **INTO** Enrollment **VALUES** (5678,'123456789',3.20);

**INSERT** **INTO** Enrollment **VALUES** (5678,'234567890',2.80);

**INSERT** **INTO** Enrollment **VALUES** (5678,'345678901',3.30);

**INSERT** **INTO** Enrollment **VALUES** (5678,'456789012',3.40);

**INSERT** **INTO** Enrollment **VALUES** (5678,'567890123',2.60);

**INSERT** **INTO** Enrollment **VALUES** (5679,'123456789',2.00);

**INSERT** **INTO** Enrollment **VALUES** (5679,'124567890',3.70);

**INSERT** **INTO** Enrollment **VALUES** (5679,'678901234',3.30);

**INSERT** **INTO** Enrollment **VALUES** (5679,'789012345',3.80);

**INSERT** **INTO** Enrollment **VALUES** (5679,'890123456',2.90);

**INSERT** **INTO** Enrollment **VALUES** (5679,'901234567',3.10);

**INSERT** **INTO** Enrollment **VALUES** (6666,'234567890',3.10);

**INSERT** **INTO** Enrollment **VALUES** (6666,'567890123',3.60);

**INSERT** **INTO** Enrollment **VALUES** (7777,'876543210',3.40);

**INSERT** **INTO** Enrollment **VALUES** (7777,'890123456',3.70);

**INSERT** **INTO** Enrollment **VALUES** (7777,'901234567',3.40);

**INSERT** **INTO** Enrollment **VALUES** (9876,'124567890',3.50);

**INSERT** **INTO** Enrollment **VALUES** (9876,'234567890',3.20);

**INSERT** **INTO** Enrollment **VALUES** (9876,'345678901',3.20);

**INSERT** **INTO** Enrollment **VALUES** (9876,'456789012',3.40);

**INSERT** **INTO** Enrollment **VALUES** (9876,'567890123',2.60);

**INSERT** **INTO** Enrollment **VALUES** (9876,'678901234',3.30);

**INSERT** **INTO** Enrollment **VALUES** (9876,'901234567',4.00);

**Problem #8 – Retrieving a subset of rows with testing for an exact string and inexact string**

Retrieve the offer number, course number, location, year, and faculty number from all course offerings in location BLM302

Retrieve the offer number, course number, location, year, and faculty number from all course offerings in location BLM 3rd floor

|  |
| --- |
| **select** OfferNo ,CourseNo ,OffLocation ,OffYear , FacNo  **from** Offering o  **where** OffLocation = 'BLM302';  **select** OfferNo ,CourseNo ,OffLocation ,OffYear , FacNo  **from** Offering o  **where** OffLocation **LIKE** 'BLM3%'; |

**Problem #9 – Using a derived column in both the column list and the WHERE clause**

Retrieve the student last name, student first name, and GPA plus 10% for all students with GPA plus 10% greater than 3

|  |
| --- |
| **select** StdLastName , StdFirstName , StdGPA \*1.10 **as** updatedGPA  **from** Student s  **where** StdGPA \*1.10 > 3; |

**Problem #10 – Retrieving the number of rows from all of our tables**

For each of our tables, retrieve the number of rows

Tables are Student, Faculty, Offering, Course, and Enrollment

(omit sorting, table aliases, and column aliases)

|  |
| --- |
| **select** **COUNT**(1) **from** Student ;  **select** **COUNT**(1) **from** Faculty ;  **select** **COUNT**(1) **from** Offering ;  **select** **COUNT**(1) **from** Course ;  **select** **COUNT**(1) **from** Enrollment ; |

**Problem #11 – Examining the effect of NULL values on aggregate functions**

Retrieve the number of rows in the Faculty table using

COUNT(\*)

COUNT(f.FacSupervisor)

How many rows does each one return? Why?

|  |
| --- |
| **select** **count**(\*) **from** Faculty f ;   * 6   **select** **count**(f.FacSupervisor) **from** Faculty f ;   * 4   There are two records in the column ‘FacSupervisor’ with NULL in the table ‘FACULTY’. So there are two faculties with no Fac Supervisors. |

**Problem #12 – Aggregates on all rows of a table**

Retrieve the average GPA for all students

|  |
| --- |
| **select** **AVG**(StdGPA)  **from** Student s ; |

**Problem #13 – Aggregates on a subset of rows of a table (using a WHERE clause)**

Retrieve the minimum GPA, maximum GPA, average GPA, and average GPA plus 10% for freshman students

|  |
| --- |
| **select** **min**(StdGPA),  **MAX**(StdGPA),  **AVG**(StdGPA),  **AVG**(StdGPA) \* 1.1  **from** Student s  **where** StdClass = 'FR'; |

**Problem #14 – Aggregates on a group of rows (using a GROUP BY clause)**

Retrieve the class name, minimum GPA, maximum GPA, average GPA, and average GPA plus 10% for each class

|  |
| --- |
| **select** StdClass, **min**(StdGPA),  **MAX**(StdGPA),  **AVG**(StdGPA),  **AVG**(StdGPA) \* 1.1 **as** updatedGPA  **from** Student s  **group** **by** StdClass ; |

**Problem #15 – Aggregates on a subset of rows that are grouped (using a WHERE clause and a GROUP BY clause)**

Retrieve the class name, minimum GPA, maximum GPA, average GPA, and average GPA plus 10% for each class but only for non-IS majors

|  |
| --- |
| **select** StdClass, **min**(StdGPA),  **MAX**(StdGPA),  **AVG**(StdGPA),  **AVG**(StdGPA) \* 1.1 **as** updatedGPA  **from** Student s  **where** StdMajor <> 'IS'  **group** **by** StdClass ; |