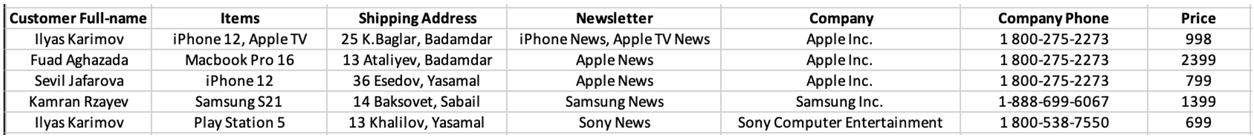
Midterm (CSCI 6441 Database Management Systems) Ilyas Karimov

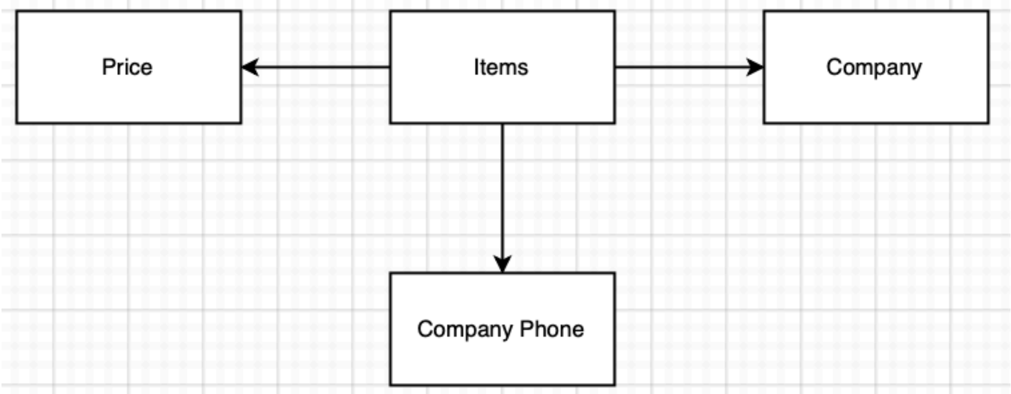
Q1. Show a single-table database that is not in 1NF and not in 2NF.
a. Show FDDs for the table.
b. Explain why it is not in 1NF and not in 2NF.
c. How does it comply with RR1 and RR2?
d. Change the database, preserving its content, into a database that is in 1NF and not 2NF.
e. Show FDDs and explain why it is in 1NF and not in 2NF.
f. Now transform it into a database that is in 1NF and 2NF.
g. Show FDDs and explain why it is in 2NF.

S1. Suppose there's an online shop where students can buy items to be delivered to their shipping address and the table down below demonstrates the database of that platform. However, it is possible for two different people to have the same name and surname. This issue also applies to my table. For example, Ilyas Karimov are two different people, and they have two different shipping addresses.



Customer Full-name	Items	Shipping Address	Newsletter	Company	Company Phone	Price
Ilyas Karimov	iPhone 12, Apple TV	25 K.Baglar, Badamdar	iPhone News, Apple TV News	Apple Inc.	1 800-275-2273	998
Fuad Aghazada	Macbook Pro 16	13 Ataliyev, Badamdar	Apple News	Apple Inc.	1 800-275-2273	2399
Sevil Jafarova	iPhone 12	36 Esedov, Yasamal	Apple News	Apple Inc.	1 800-275-2273	799
Kamran Rzayev	Samsung S21	14 Baksovet, Sabail	Samsung News	Samsung Inc.	1-888-699-6067	1399
Ilyas Karimov	Play Station 5	13 Khalilov, Yasamal	Sony News	Sony Computer Entertainment	1 800-538-7550	699

a. FDDs: Item→ Company, Item→ Company Phone, Item→ Price.

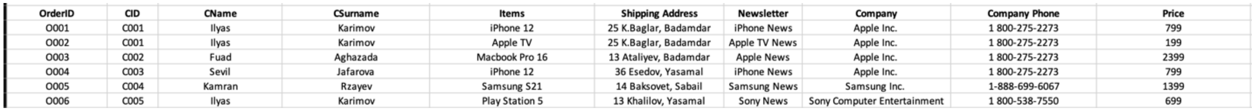


Note: 1. There are two different people named Ilyas Karimov, hence Customer Full-name table will not have any dependence thinking that the same value is repeated.
2. Imagine the student live in the same apartment, therefore have the same shipping address, which means Shipping Address→ Customer Full-name is not possible.

b. It is not in 1NF because: 1. Item attribute is not single-valued, 2. Rows are not uniquely identified, there is a need of **CustomerID** and **OrderID** attributes
It is not in 2NF because: 1. Not all attributes are dependent on the key, such as **Price, Company** attributes themselves.

c. RR1: Each table **exactly** describes **one** entity type, not two or more of them. The rule is violated because **Company**, **Company Phone** and **Price** attributes are rather about the Item, not about the **Customer**.
RR2: Each fact is represented **only once** in the database. The rule is violated by the **Company** and **Company Phone** attributes.

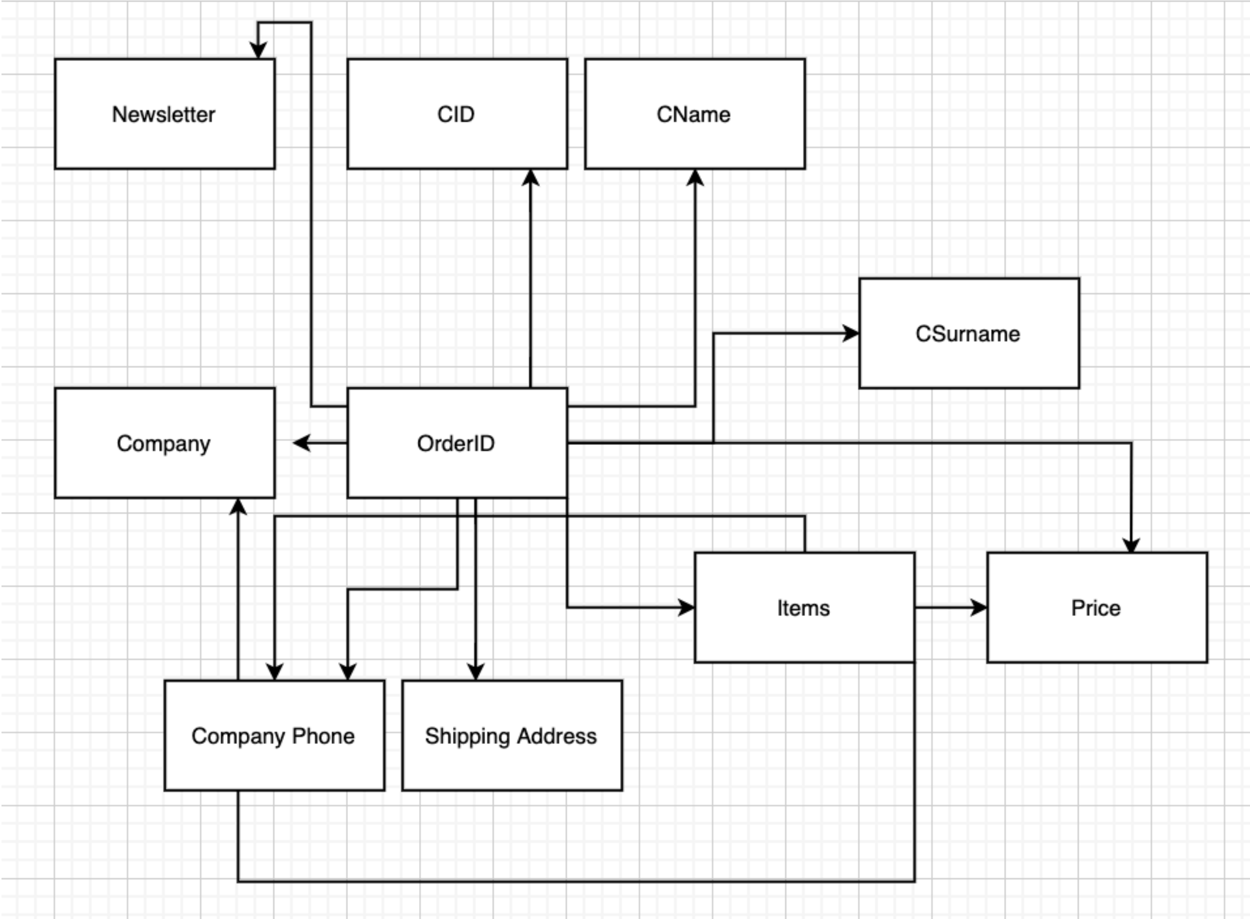
d. The table is in 1NF, not in 2NF.
Note: keeping name and surname in different attributes is optional.



OrderID	CID	CName	CSurname	Items	Shipping Address	Newsletter	Company	Company Phone	Price
O001	C001	Ilyas	Karimov	iPhone 12	25 K.Baglar, Badamdar	iPhone News	Apple Inc.	1 800-275-2273	799
O002	C001	Ilyas	Karimov	Apple TV	25 K.Baglar, Badamdar	Apple TV News	Apple Inc.	1 800-275-2273	199
O003	C002	Fuad	Aghazada	Macbook Pro 16	13 Ataliyev, Badamdar	Apple News	Apple Inc.	1 800-275-2273	2399
O004	C003	Sevil	Jafarova	iPhone 12	36 Esedov, Yasamal	iPhone News	Apple Inc.	1 800-275-2273	799
O005	C004	Kamran	Rzayev	Samsung S21	14 Baksovet, Sabail	Samsung News	Samsung Inc.	1-888-699-6067	1399
O006	C005	Ilyas	Karimov	Play Station 5	13 Khalilov, Yasamal	Sony News	Sony Computer Entertainment	1 800-538-7550	699

e. FDDs: OrderID→CID, OrderID→CName, OrderID→CSurname, OrderID→Items, OrderID→Shipping Address, OrderID→Newsletter, OrderID→Company, OrderID→Company Phone, OrderID→Price, CID→ CName, CID→CSurname, CID→Shipping Address, Item→ Company, Item→ Company Phone, Item→ Price.

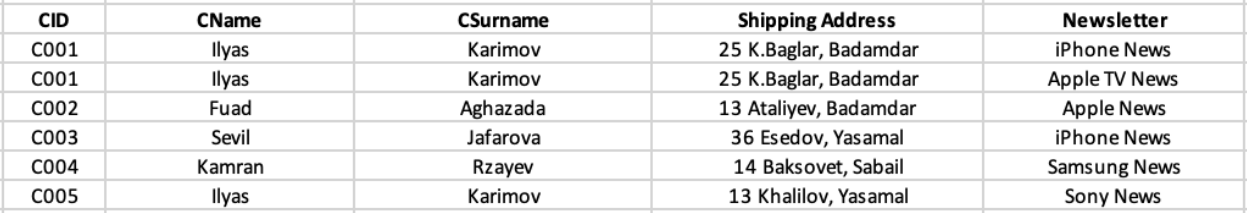
f.



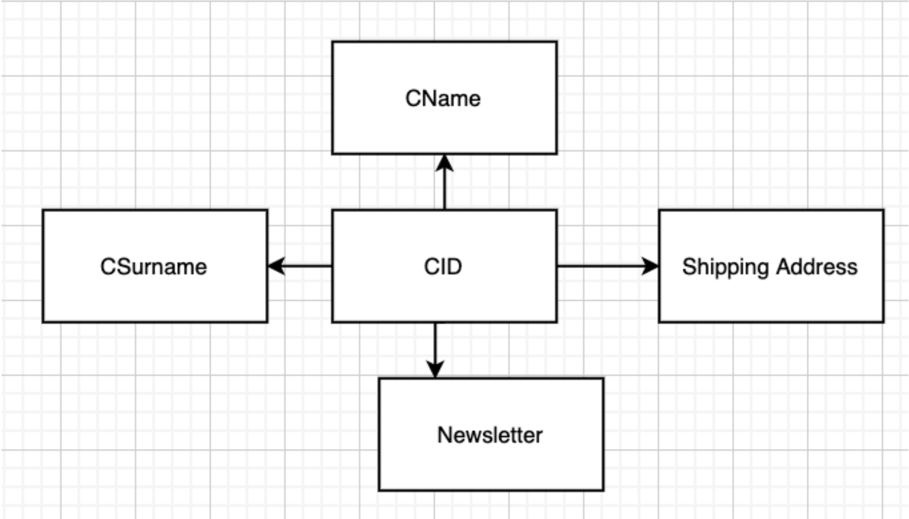
The table is in 1NF because: 1. Entries are the same type, 2. Rows are uniquely identified, 3. Each cell is single-valued.

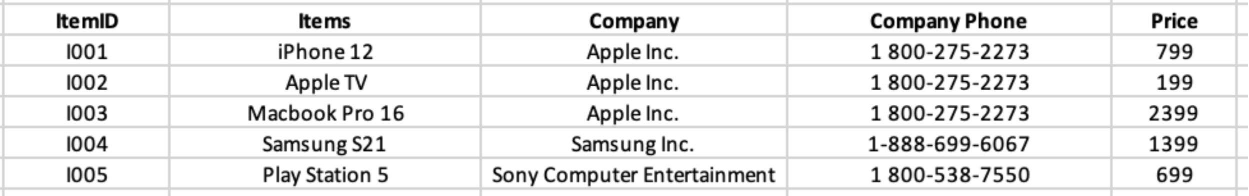
The table is not in 2NF because: 1. Not all attributes are dependent on the key, such as **Price, Company** attributes themselves are not dependent on the **CID**.

g. The tables are in 2NF because all attributes depend on the key now and it is in 1NF.

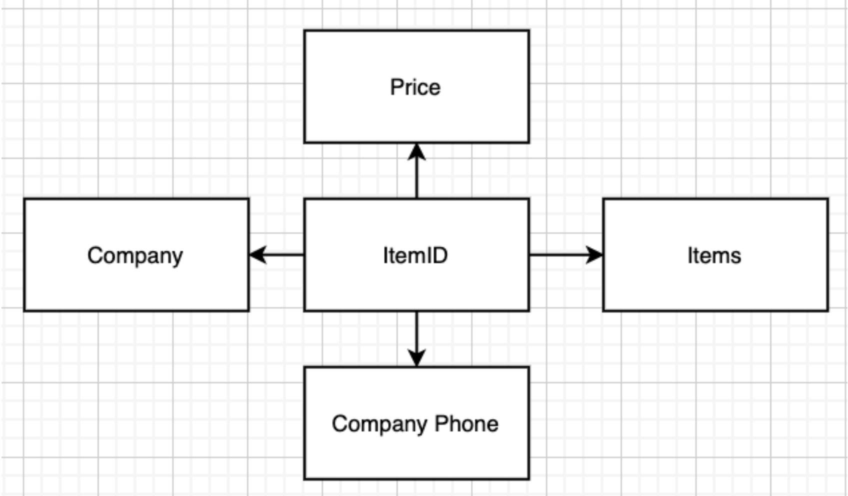


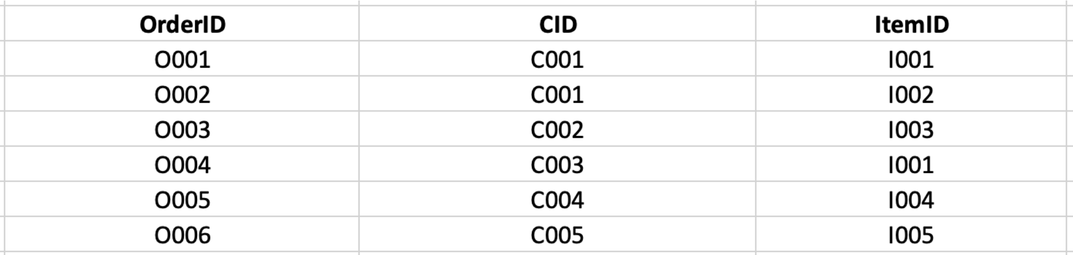
CID	CName	CSurname	Shipping Address	Newsletter
C001	Ilyas	Karimov	25 K.Baglar, Badamdar	iPhone News
C001	Ilyas	Karimov	25 K.Baglar, Badamdar	Apple TV News
C002	Fuad	Aghazada	13 Ataliyev, Badamdar	Apple News
C003	Sevil	Jafarova	36 Esedov, Yasamal	iPhone News
C004	Kamran	Rzayev	14 Baksovet, Sabail	Samsung News
C005	Ilyas	Karimov	13 Khalilov, Yasamal	Sony News



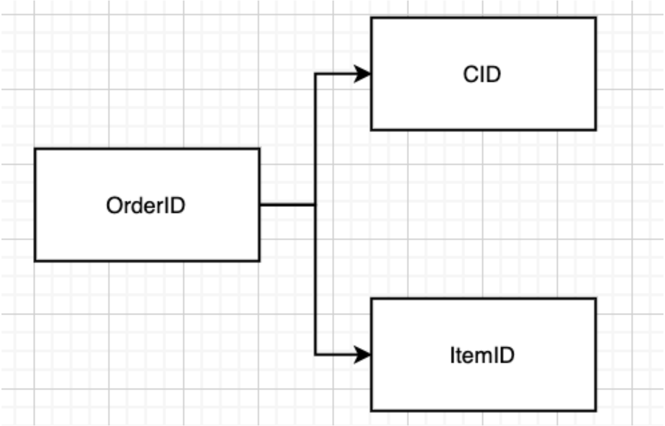


ItemID	Items	Company	Company Phone	Price
I001	iPhone 12	Apple Inc.	1 800-275-2273	799
I002	Apple TV	Apple Inc.	1 800-275-2273	199
I003	Macbook Pro 16	Apple Inc.	1 800-275-2273	2399
I004	Samsung S21	Samsung Inc.	1-888-699-6067	1399
I005	Play Station 5	Sony Computer Entertainment	1 800-538-7550	699



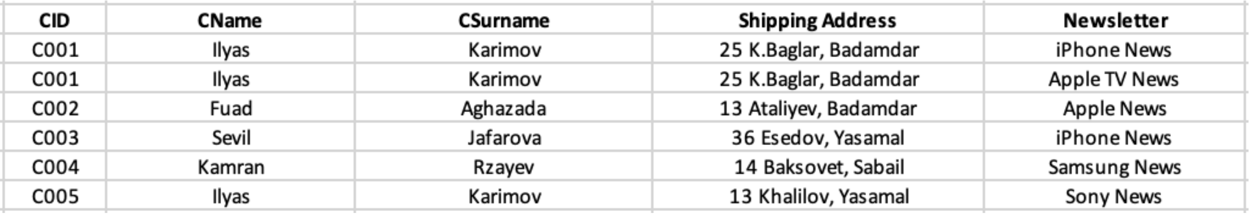


OrderID	CID	ItemID
O001	C001	I001
O002	C001	I002
O003	C002	I003
O004	C003	I001
O005	C004	I004
O006	C005	I005

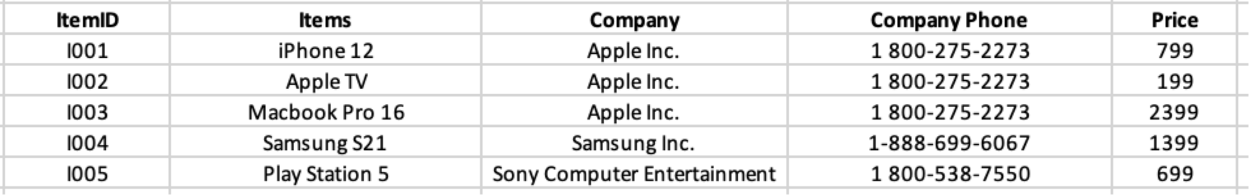


Q2. Show a single-table database that is not in 3NF and not in 4NF.
a. Show FDDs for the table.
b. Explain why it is not in 3NF and not in 4NF.
c. How does it comply with RR1 and RR2?
d. Now transform it into a new database with the same content that is in 3NF and not 4NF.
e. Show FDDs and explain why it is in 3NF and not 4NF.
f. Now transform it into a database that is in 3NF and 4NF.
g. Show FDDs and explain why it is in 4NF.

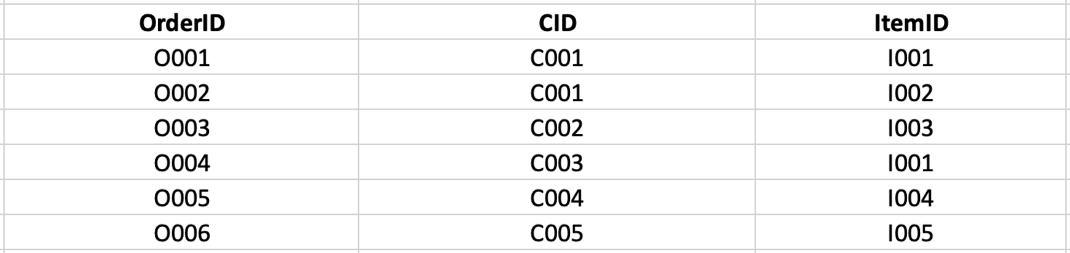
S2.



CID	CName	CSurname	Shipping Address	Newsletter
C001	Ilyas	Karimov	25 K.Baglar, Badamdar	iPhone News
C001	Ilyas	Karimov	25 K.Baglar, Badamdar	Apple TV News
C002	Fuad	Aghazada	13 Ataliyev, Badamdar	Apple News
C003	Sevil	Jafarova	36 Esedov, Yasamal	iPhone News
C004	Kamran	Rzayev	14 Baksovet, Sabail	Samsung News
C005	Ilyas	Karimov	13 Khalilov, Yasamal	Sony News



ItemID	Items	Company	Company Phone	Price
I001	iPhone 12	Apple Inc.	1 800-275-2273	799
I002	Apple TV	Apple Inc.	1 800-275-2273	199
I003	Macbook Pro 16	Apple Inc.	1 800-275-2273	2399
I004	Samsung S21	Samsung Inc.	1-888-699-6067	1399
I005	Play Station 5	Sony Computer Entertainment	1 800-538-7550	699



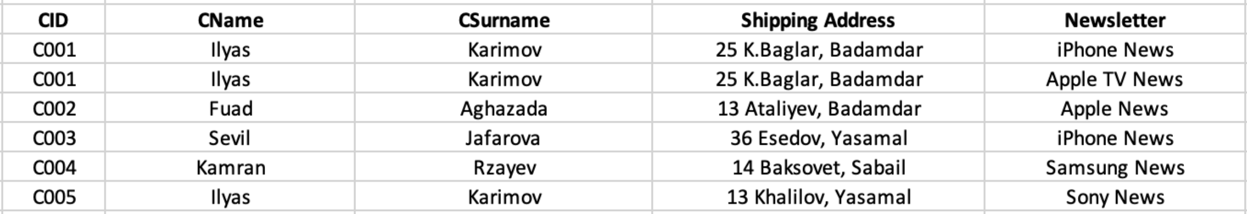
OrderID	CID	ItemID
O001	C001	I001
O002	C001	I002
O003	C002	I003
O004	C003	I001
O005	C004	I004
O006	C005	I005

a. The Dependencies are the same as above in 1.g.

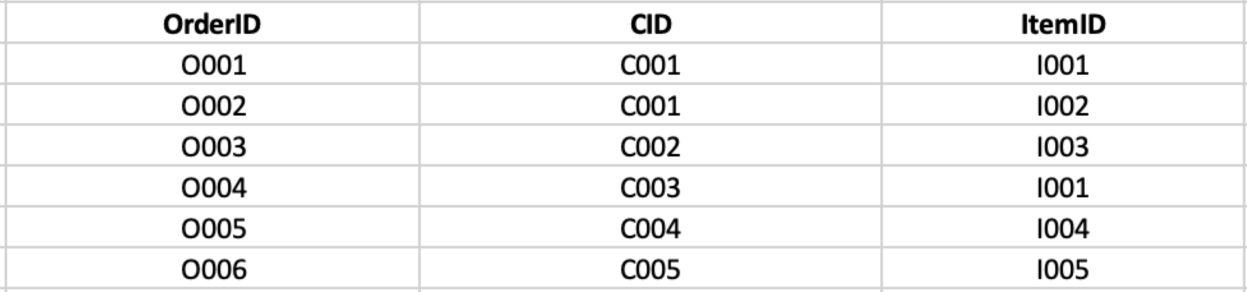
b. It is not in 3NF because some columns can be determined by non-key column, such as **Company** can determine **Company phone**.
It is not in 4NF because there’s a multi-valued dependence, as the person with CID C001 has subscribed to Newsletters iPhone and Apple TV, and suppose he unsubscribes Apple TV news, the row should be erased… What if he unsubscribes from both of them, what now? Erasing him from the table doesn’t seem be the solution.

c. RR1: Each table **exactly** describes **one** entity type, not two or more of them. The rule is complied.
RR2: Each fact is represented **only once** in the database. This rule is violated because for C001 the Shipping Address, CName fact is repeated.

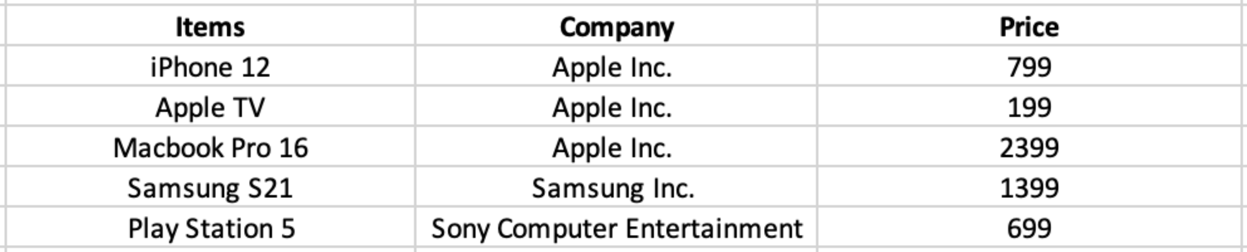
d.



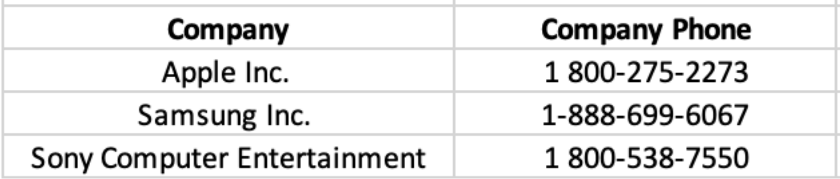
CID	CName	CSurname	Shipping Address	Newsletter
C001	Ilyas	Karimov	25 K.Baglar, Badamdar	iPhone News
C001	Ilyas	Karimov	25 K.Baglar, Badamdar	Apple TV News
C002	Fuad	Aghazada	13 Ataliyev, Badamdar	Apple News
C003	Sevil	Jafarova	36 Esedov, Yasamal	iPhone News
C004	Kamran	Rzayev	14 Baksovet, Sabail	Samsung News
C005	Ilyas	Karimov	13 Khalilov, Yasamal	Sony News



OrderID	CID	ItemID
O001	C001	I001
O002	C001	I002
O003	C002	I003
O004	C003	I001
O005	C004	I004
O006	C005	I005



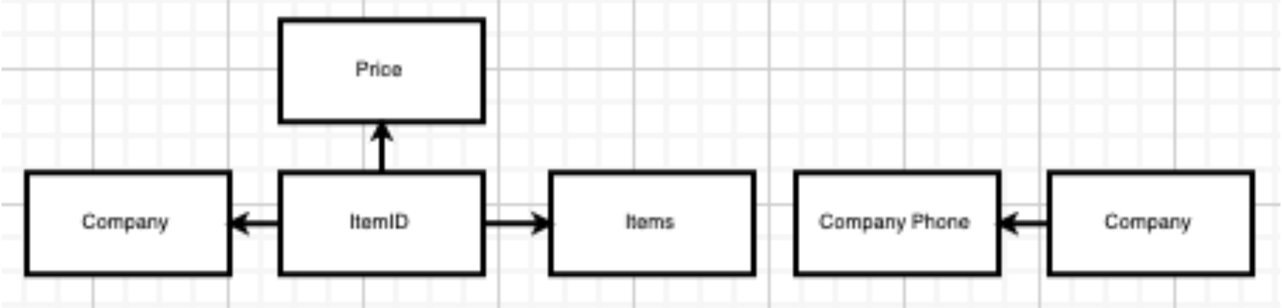
Items	Company	Price
iPhone 12	Apple Inc.	799
Apple TV	Apple Inc.	199
Macbook Pro 16	Apple Inc.	2399
Samsung S21	Samsung Inc.	1399
Play Station 5	Sony Computer Entertainment	699



Company	Company Phone
Apple Inc.	1 800-275-2273
Samsung Inc.	1-888-699-6067
Sony Computer Entertainment	1 800-538-7550

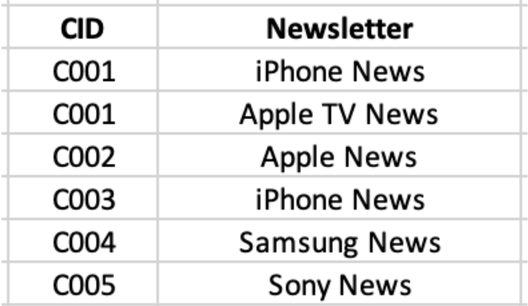
e.

FDD:

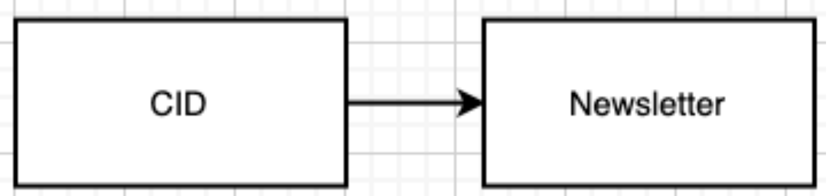


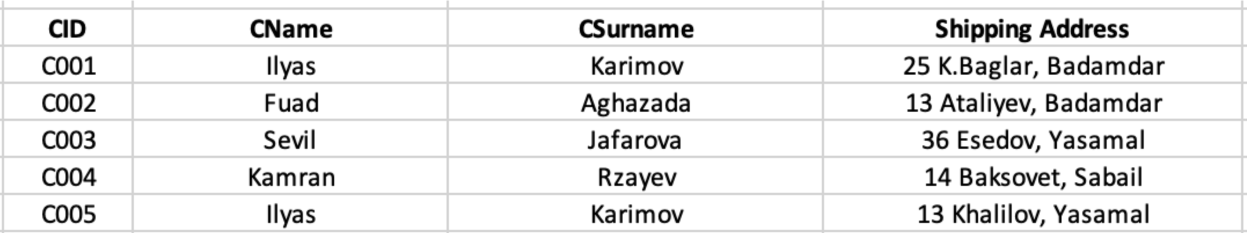
The table is in 3NF because there is no transitive dependence but not in 4NF because there's a multi-valued dependence.

f.

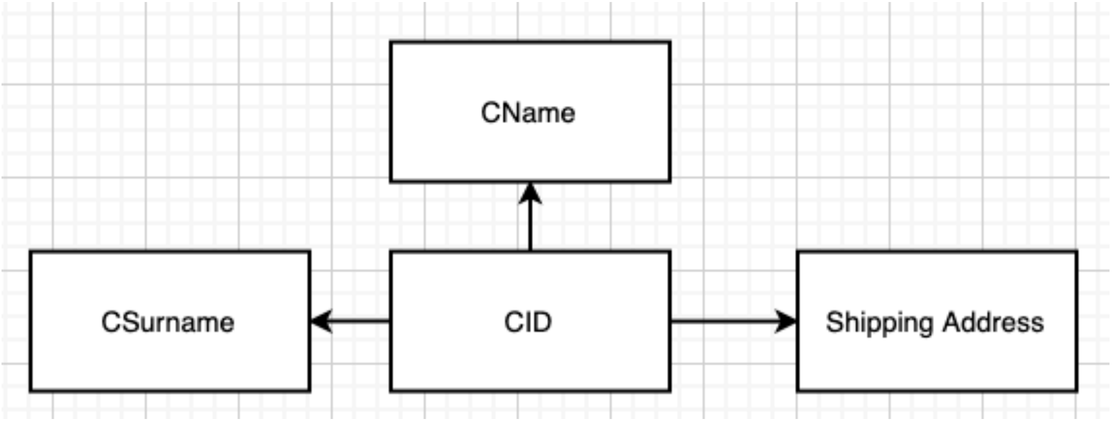


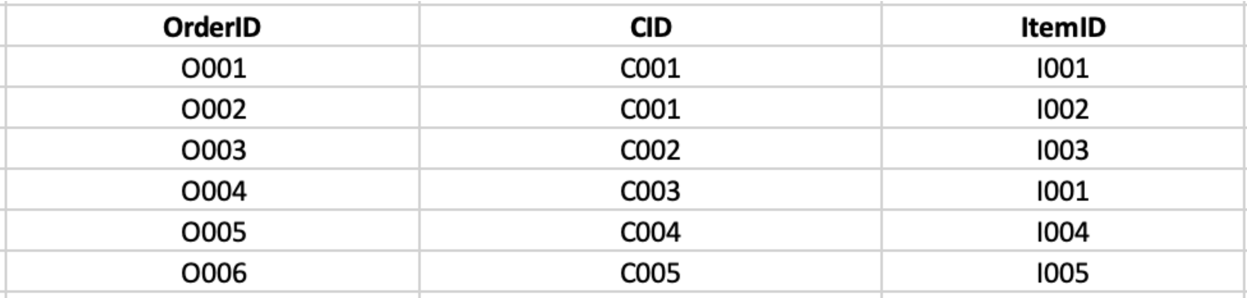
CID	Newsletter
C001	iPhone News
C001	Apple TV News
C002	Apple News
C003	iPhone News
C004	Samsung News
C005	Sony News



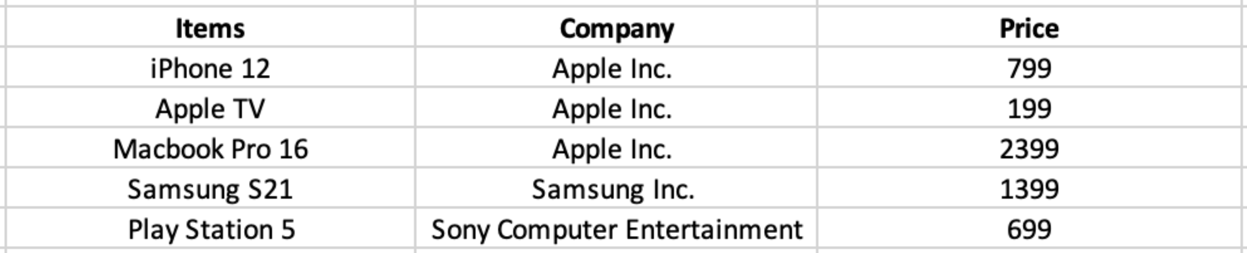


CID	CName	CSurname	Shipping Address
C001	Ilyas	Karimov	25 K.Baglar, Badamdar
C002	Fuad	Aghazada	13 Ataliyev, Badamdar
C003	Sevil	Jafarova	36 Esedov, Yasamal
C004	Kamran	Rzayev	14 Baksovet, Sabail
C005	Ilyas	Karimov	13 Khalilov, Yasamal

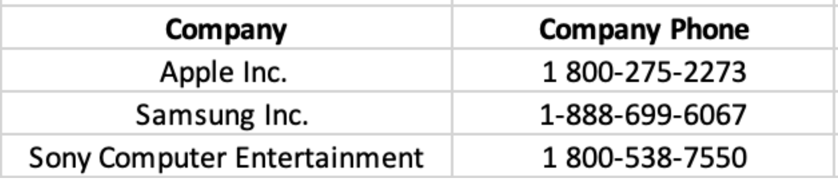




OrderID	CID	ItemID
O001	C001	I001
O002	C001	I002
O003	C002	I003
O004	C003	I001
O005	C004	I004
O006	C005	I005

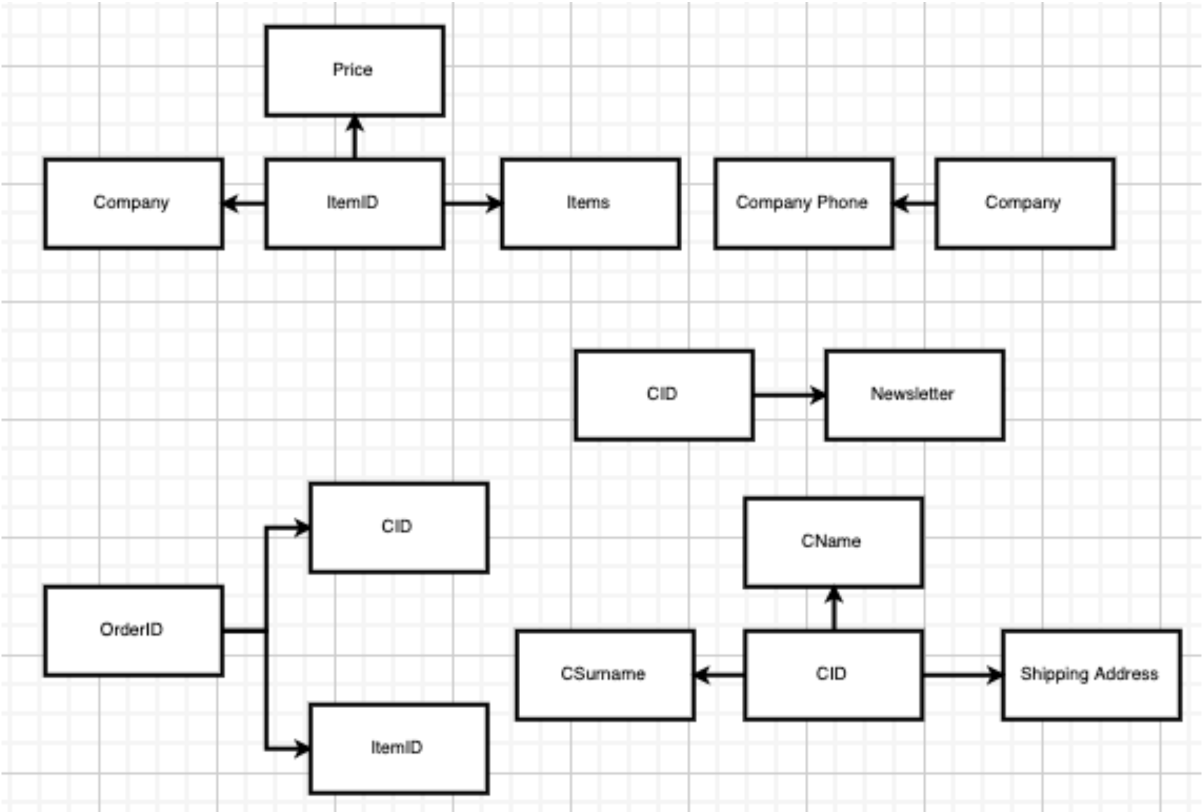


Items	Company	Price
iPhone 12	Apple Inc.	799
Apple TV	Apple Inc.	199
Macbook Pro 16	Apple Inc.	2399
Samsung S21	Samsung Inc.	1399
Play Station 5	Sony Computer Entertainment	699



Company	Company Phone
Apple Inc.	1 800-275-2273
Samsung Inc.	1-888-699-6067
Sony Computer Entertainment	1 800-538-7550

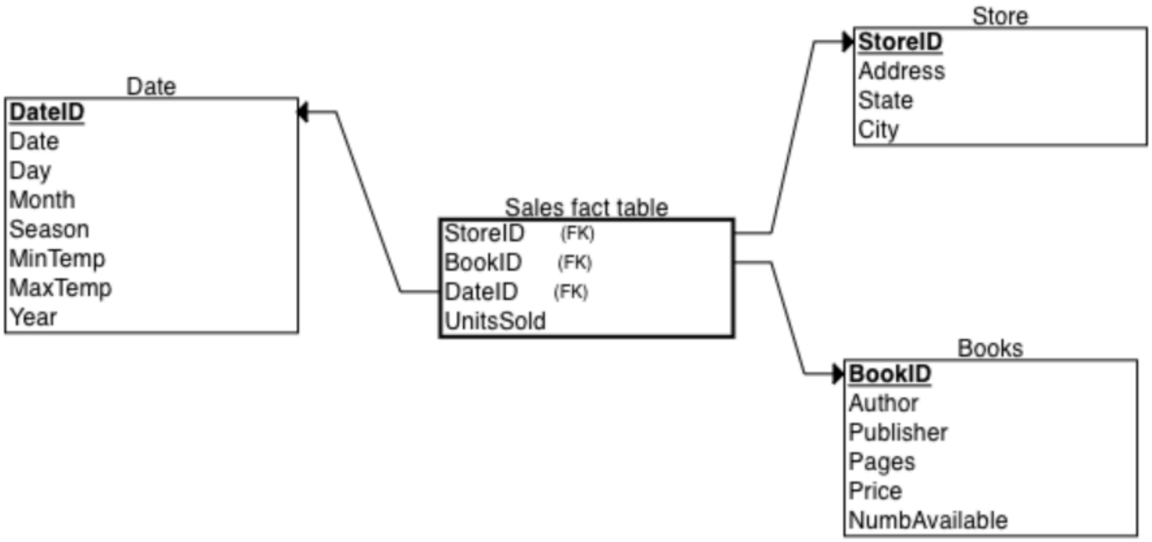
Final FDDs:



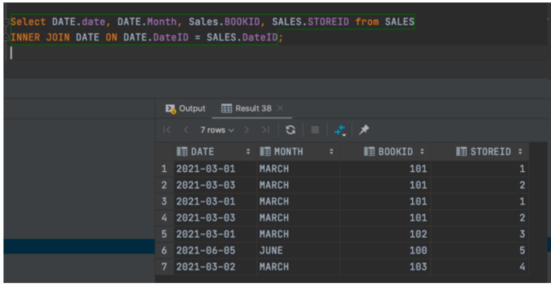
Q3. Consider a retail store selling books. Sales are made to customers who come to the store and purchase in person. Sales are also made on the store&apos;s website; these sales are shipped to customers. The store is a chain, that has branches in 24 cities in England, and is growing rapidly. They would like to establish a data analysis operation. You have been given a contract to develop the data model for their on-line analytical processing application. Design a star schema for them that will allow them to analyze sales by book publisher, by day of the week, month of the year, season, range of outdoor temperature, weather, day of the week, month, number of pages and book price.

Put ten rows of sample data into the fact table and enough rows to your dimension tables to show the operation of the star schema. Construct a few SELECT statements that show several of the types of analysis given above.

S3. Below is the star scheme of this database.



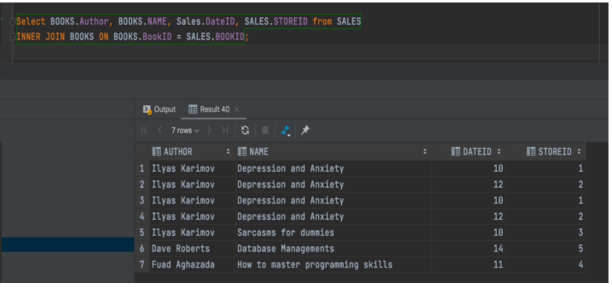
Showing the month of the purchases.



```
Select DATE.date, DATE.Month, Sales.BOOKID, SALES.STOREID from SALES
INNER JOIN DATE ON DATE.DateID = SALES.DateID;
```

```
     DATE        MONTH    BOOKID   STOREID
1    2021-03-01  MARCH       101         1
2    2021-03-03  MARCH       101         2
3    2021-03-01  MARCH       101         1
4    2021-03-03  MARCH       101         2
5    2021-03-01  MARCH       102         3
6    2021-06-05  JUNE        100         5
7    2021-03-02  MARCH       103         4
```

Showing the authors and names of the books



```
Select BOOKS.Author, BOOKS.NAME, Sales.DateID, SALES.STOREID from SALES
INNER JOIN BOOKS ON BOOKS.BookID = SALES.BOOKID;
```

```
     AUTHOR          NAME                              DATEID   STOREID
1    Ilyas Karimov   Depression and Anxiety                10         1
2    Ilyas Karimov   Depression and Anxiety                12         2
3    Ilyas Karimov   Depression and Anxiety                10         1
4    Ilyas Karimov   Depression and Anxiety                12         2
5    Ilyas Karimov   Sarcasms for dummies                  10         3
6    Dave Roberts    Database Managements                  14         5
7    Fuad Aghazada   How to master programming skills      11         4
```

```
Select DATE.date, DATE.Month, Sales.DateID, SALES.STOREID from SALES
INNER JOIN DATE ON DATE.DateID = SALES.DateID;
```

```
Select BOOKS.Author, BOOKS.NAME, Sales.DateID, SALES.STOREID from SALES
INNER JOIN BOOKS ON BOOKS.BookID = SALES.BOOKID;
```

Note: Check Appendix 3. to see all queries.

Q4. Roberts's Rule One and Roberts's Rule Two are equivalent if and only if they define the same set of databases. That is, if every database that satisfies Rule One also satisfies Rule Two. Are they equivalent? Prove your answer.

S4. I would say that they are correlated but definitely not the same. What I mean by correlated is that, for example, we have a table, say Employee table, and if this table belongs to only one entity, that is, if the attributes of the Department are not in this table, then it will help to avoid repeating the same facts. In this case, we can say that to comply with Roberts's Rule 1 means to comply with Rule 2.
Now suppose, there is a small company with 10 departments and one person from each department, i.e., a total of 10 people work, and each person is from a different department. Now

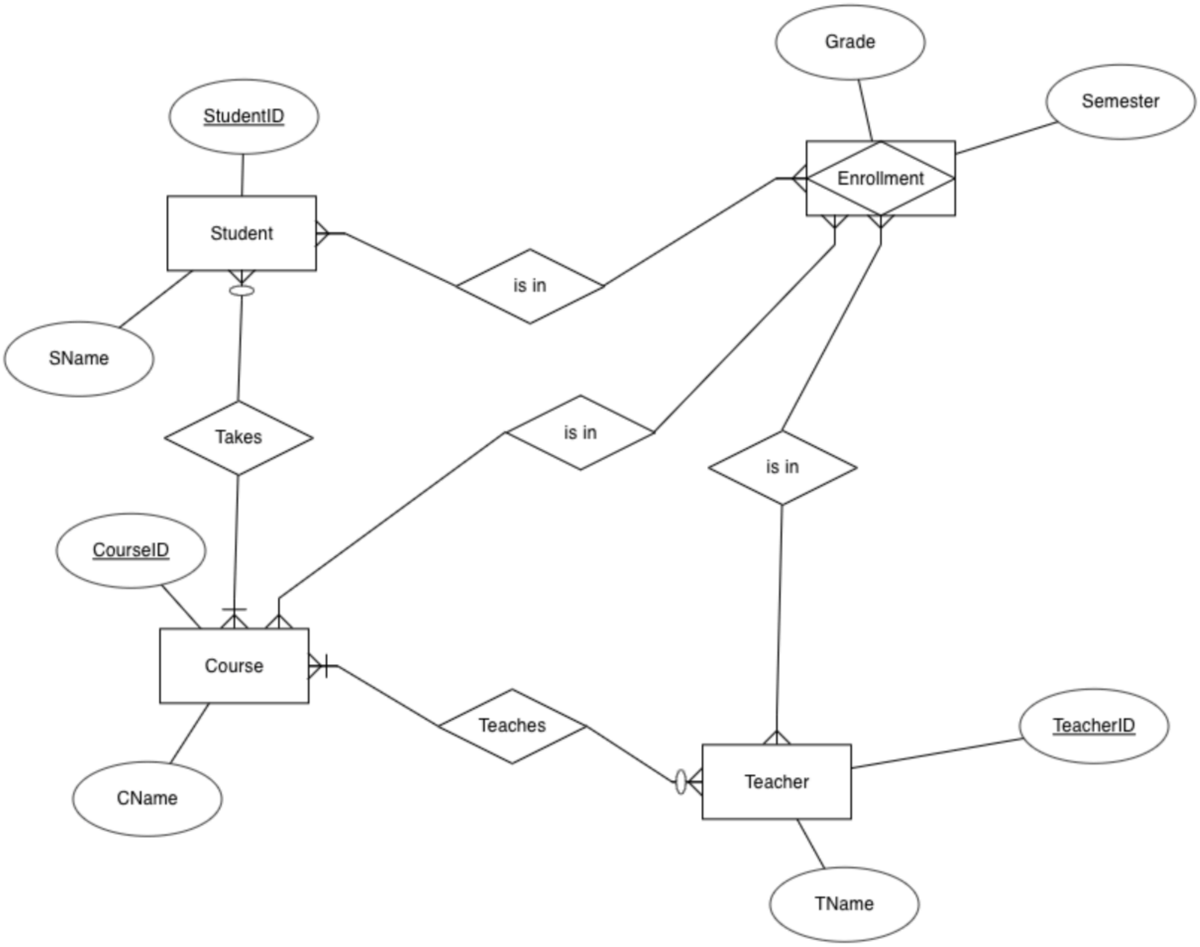
suppose an Employee table whose attributes are EMP #, ENAME, JOB, DEPTNO, DEPTNAME. This 10-row table doesn't actually repeat any facts, but that doesn't mean it fits Rule 1, actually, this table model **violates** Roberts's Rule 1. Which proves that those rules are not the same rules.

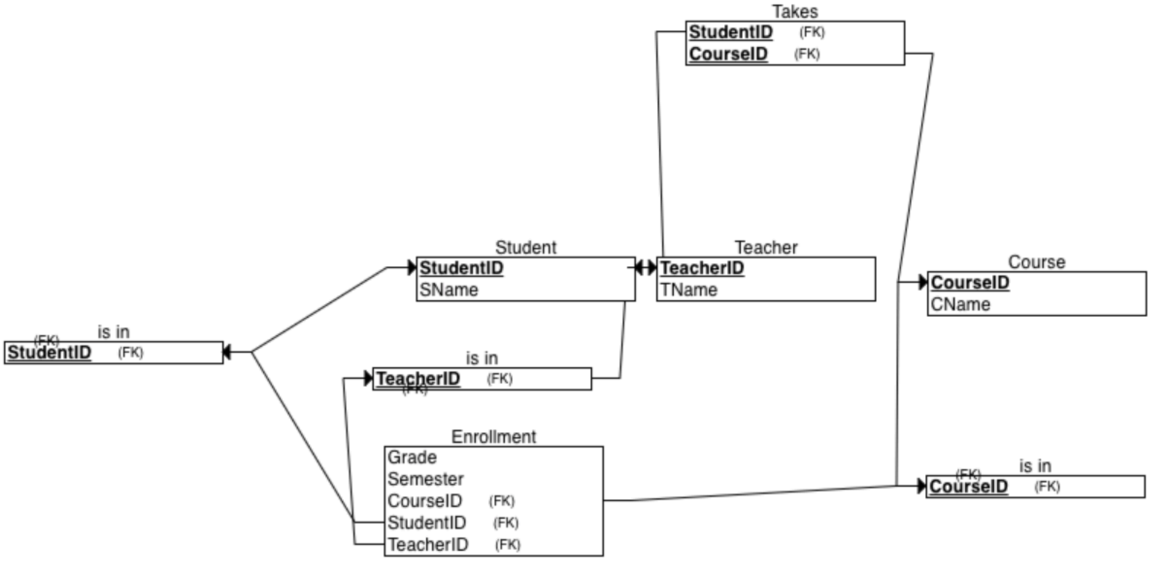
Q5. Design a data model for the problem below.

Show a Chen-style ERD, FDDs, and then convert it to relational tables and show a crows-foot ERD for the relational tables.

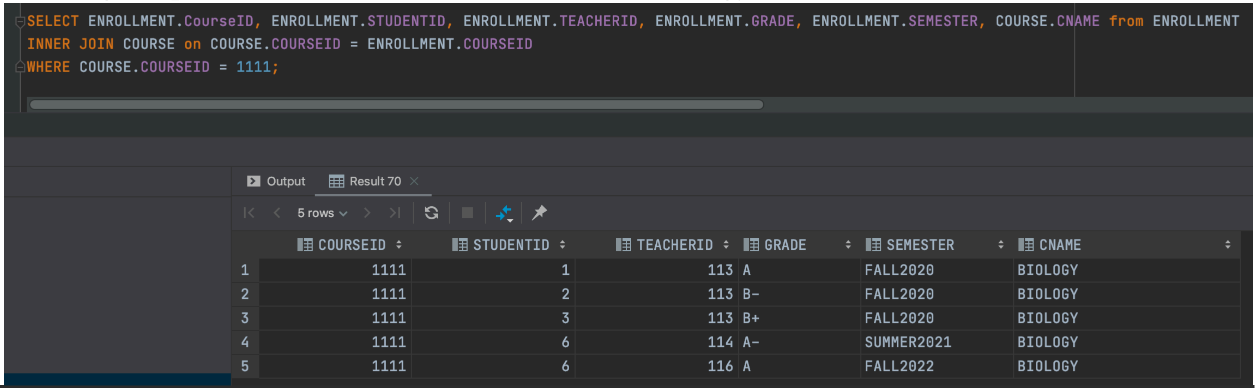
Be sure to show cardinality and optionality on all of your diagrams.
Avoid the use of **generated keys** in your Chen diagram wherever you can.
A school has students who each take courses, teachers who each teach courses. We want to be able to report grades of one student, a table of all student grades in one semester, a report on all grades for a course, a report on all grades given by a single teacher.
Put ten teachers, ten students and ten courses into your database, and show these reports.

S5.
Many students can take many courses and student's taking a course is a must, but course is still a course without students taking them. The same applies to Teachers and Courses. The Enrollment





Showing People who have taken the course 1111 (Biology)

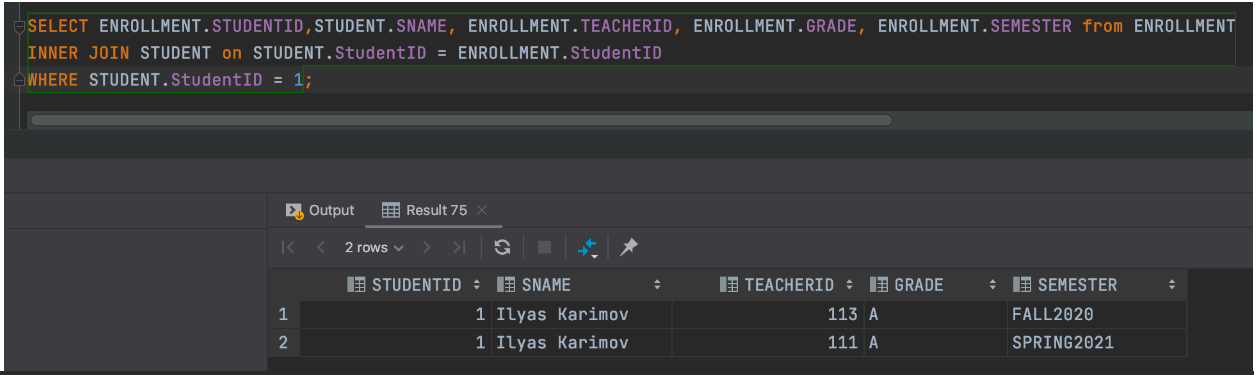


```
SELECT ENROLLMENT.CourseID, ENROLLMENT.STUDENTID, ENROLLMENT.TEACHERID, ENROLLMENT.GRADE, ENROLLMENT.SEMESTER, COURSE.CNAME from ENROLLMENT
INNER JOIN COURSE on COURSE.COURSEID = ENROLLMENT.COURSEID
WHERE COURSE.COURSEID = 1111;
```

	COURSEID	STUDENTID	TEACHERID	GRADE	SEMESTER	CNAME
1	1111	1	113	A	FALL2020	BIOLOGY
2	1111	2	113	B-	FALL2020	BIOLOGY
3	1111	3	113	B+	FALL2020	BIOLOGY
4	1111	6	114	A-	SUMMER2021	BIOLOGY
5	1111	6	116	A	FALL2022	BIOLOGY

```
SELECT ENROLLMENT.CourseID, ENROLLMENT.STUDENTID, ENROLLMENT.TEACHERID,
ENROLLMENT.GRADE, ENROLLMENT.SEMESTER, COURSE.CNAME from ENROLLMENT
INNER JOIN COURSE on COURSE.COURSEID = ENROLLMENT.COURSEID
WHERE COURSE.COURSEID = 1111;
```

Showing student with StudentID 1 who have taken courses.



```
SELECT ENROLLMENT.STUDENTID,STUDENT.SNAME, ENROLLMENT.TEACHERID, ENROLLMENT.GRADE, ENROLLMENT.SEMESTER from ENROLLMENT
INNER JOIN STUDENT on STUDENT.StudentID = ENROLLMENT.StudentID
WHERE STUDENT.StudentID = 1;
```

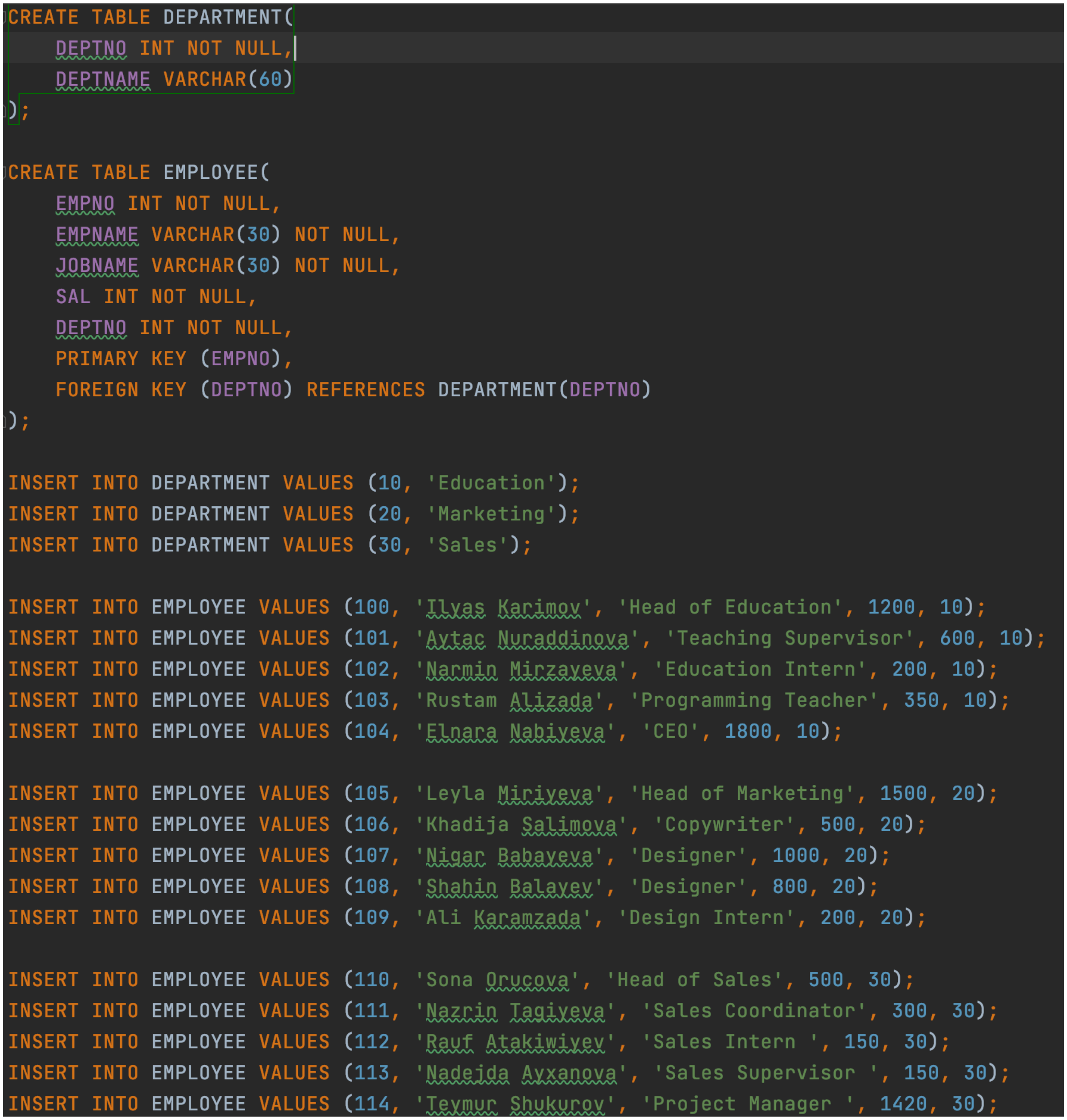
	STUDENTID	SNAME	TEACHERID	GRADE	SEMESTER
1	1	Ilyas Karimov	113	A	FALL2020
2	1	Ilyas Karimov	111	A	SPRING2021

```
SELECT ENROLLMENT.STUDENTID,STUDENT.SNAME, ENROLLMENT.TEACHERID,
ENROLLMENT.GRADE, ENROLLMENT.SEMESTER from ENROLLMENT
```

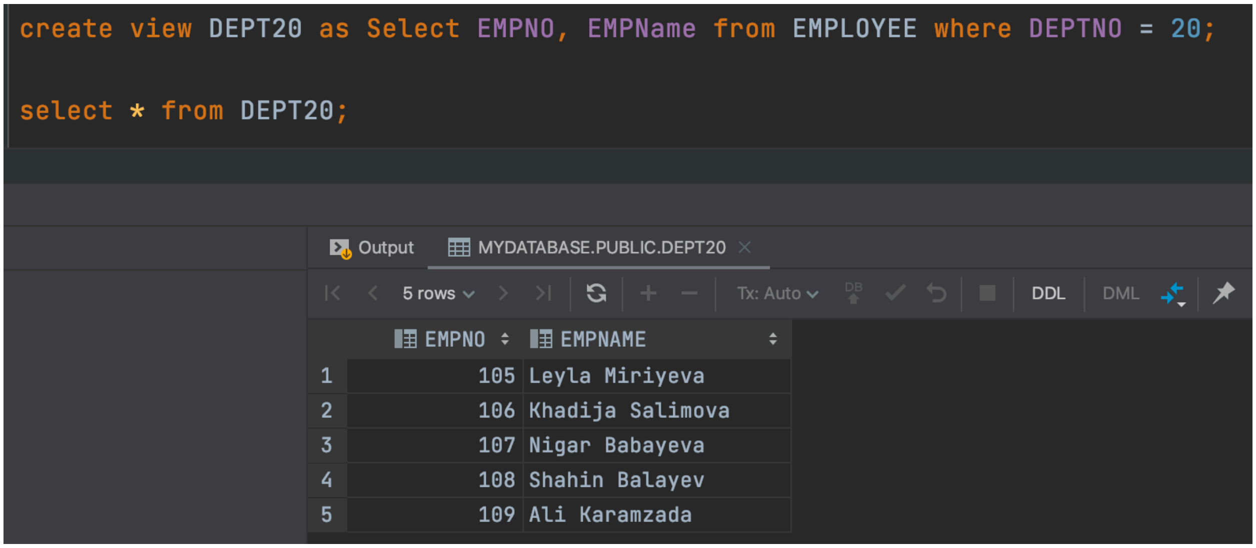
```
INNER JOIN STUDENT on STUDENT.StudentID = ENROLLMENT.StudentID
WHERE STUDENT.StudentID = 1;
```

Q6. Create a department table like the ones discussed in class, with departments 10, 20, and 30, each with 5 employees. Now create a view called DEPT20 that shows only the names and employee numbers for employees in department 20. Demonstrate the query. Now change the name of one employee in department 20 by updating the view. Now define another view on the same table that cannot be updated, and demonstrate that it can&apos;t be updated. Explain why one view was updateable and the other was not. What is the underlying principle that determines when a view is update-able? For these two views, attempt to insert a new row into the view; show and explain the result.

S6.



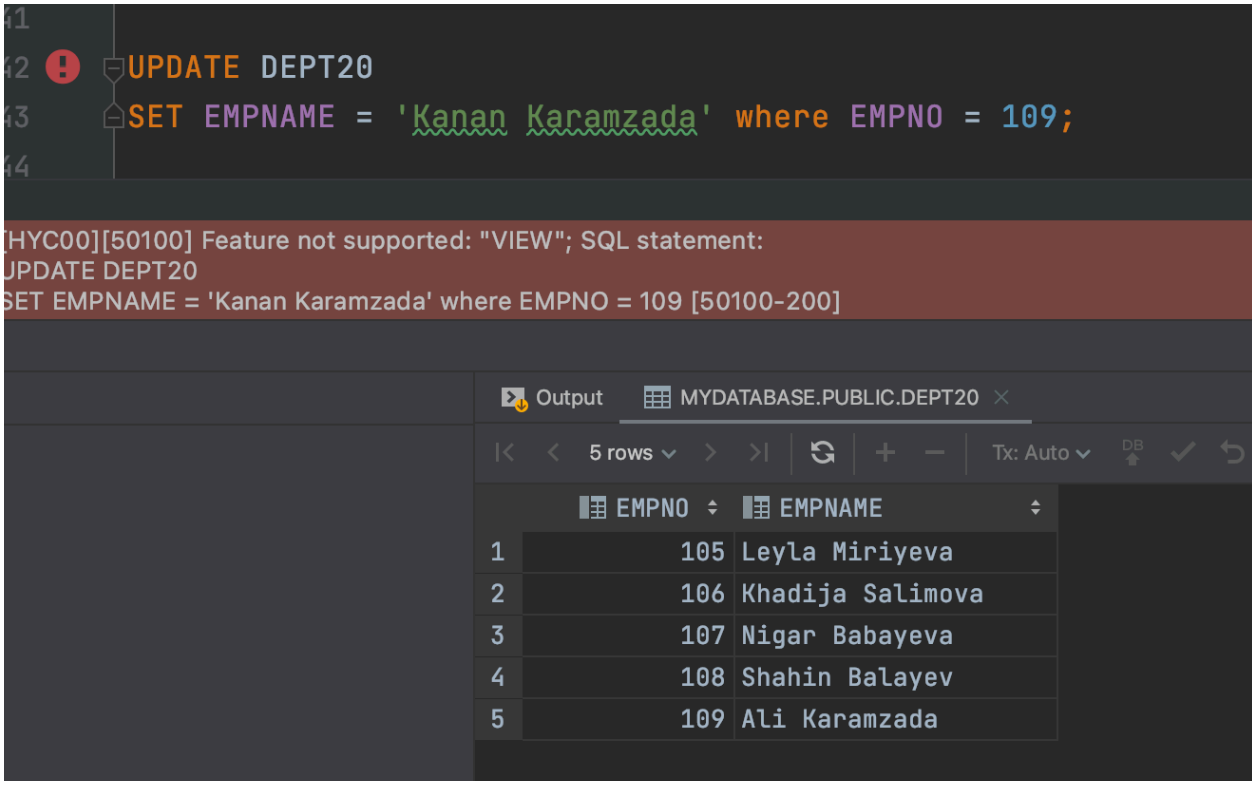
```sql
CREATE TABLE DEPARTMENT(
    DEPTNO INT NOT NULL,
    DEPTNAME VARCHAR(60)
);

CREATE TABLE EMPLOYEE(
    EMPNO INT NOT NULL,
    EMPNAME VARCHAR(30) NOT NULL,
    JOBNAME VARCHAR(30) NOT NULL,
    SAL INT NOT NULL,
    DEPTNO INT NOT NULL,
    PRIMARY KEY (EMPNO),
    FOREIGN KEY (DEPTNO) REFERENCES DEPARTMENT(DEPTNO)
);

INSERT INTO DEPARTMENT VALUES (10, 'Education');
INSERT INTO DEPARTMENT VALUES (20, 'Marketing');
INSERT INTO DEPARTMENT VALUES (30, 'Sales');

INSERT INTO EMPLOYEE VALUES (100, 'Ilyas Karimov', 'Head of Education', 1200, 10);
INSERT INTO EMPLOYEE VALUES (101, 'Aytac Nuraddinova', 'Teaching Supervisor', 600, 10);
INSERT INTO EMPLOYEE VALUES (102, 'Narmin Mirzayeva', 'Education Intern', 200, 10);
INSERT INTO EMPLOYEE VALUES (103, 'Rustam Alizada', 'Programming Teacher', 350, 10);
INSERT INTO EMPLOYEE VALUES (104, 'Elnara Nabiyeva', 'CEO', 1800, 10);

INSERT INTO EMPLOYEE VALUES (105, 'Leyla Miriyeva', 'Head of Marketing', 1500, 20);
INSERT INTO EMPLOYEE VALUES (106, 'Khadija Salimova', 'Copywriter', 500, 20);
INSERT INTO EMPLOYEE VALUES (107, 'Nigar Babayeva', 'Designer', 1000, 20);
INSERT INTO EMPLOYEE VALUES (108, 'Shahin Balayev', 'Designer', 800, 20);
INSERT INTO EMPLOYEE VALUES (109, 'Ali Karamzada', 'Design Intern', 200, 20);

INSERT INTO EMPLOYEE VALUES (110, 'Sona Orucova', 'Head of Sales', 500, 30);
INSERT INTO EMPLOYEE VALUES (111, 'Nazrin Tagiyeva', 'Sales Coordinator', 300, 30);
INSERT INTO EMPLOYEE VALUES (112, 'Rauf Atakiwiyev', 'Sales Intern ', 150, 30);
INSERT INTO EMPLOYEE VALUES (113, 'Nadejda Ayxanova', 'Sales Supervisor ', 150, 30);
INSERT INTO EMPLOYEE VALUES (114, 'Teymur Shukurov', 'Project Manager ', 1420, 30);
```



```
create view DEPT20 as Select EMPNO, EMPName from EMPLOYEE where DEPTNO = 20;

select * from DEPT20;
```

```
   EMPNO  EMPNAME
1    105  Leyla Miriyeva
2    106  Khadija Salimova
3    107  Nigar Babayeva
4    108  Shahin Balayev
5    109  Ali Karamzada
```

I’m using DataGrip and views are not updateable for H2 but for Oracle, this update must have worked.

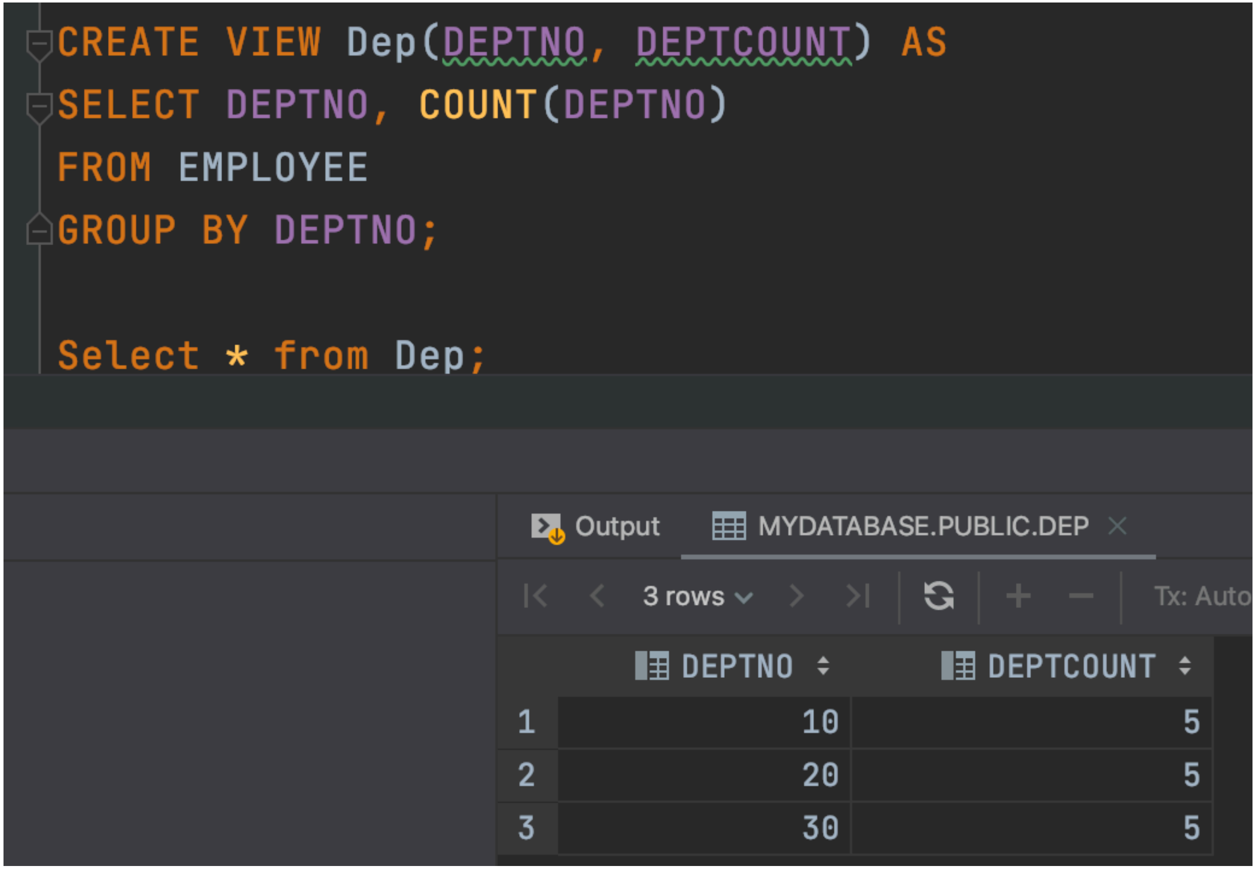


```
UPDATE DEPT20
SET EMPNAME = 'Kanan Karamzada' where EMPNO = 109;
```

```
[HYC00][50100] Feature not supported: "VIEW"; SQL statement:
UPDATE DEPT20
SET EMPNAME = 'Kanan Karamzada' where EMPNO = 109 [50100-200]
```

```
   EMPNO  EMPNAME
1    105  Leyla Miriyeva
2    106  Khadija Salimova
3    107  Nigar Babayeva
4    108  Shahin Balayev
5    109  Ali Karamzada
```

A view is non-updatable as Group By and Aggregate function Count has been used in this view.



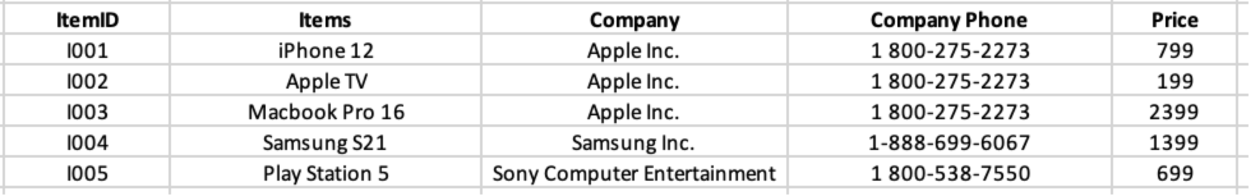
```
CREATE VIEW Dep(DEPTNO, DEPTCOUNT) AS
SELECT DEPTNO, COUNT(DEPTNO)
FROM EMPLOYEE
GROUP BY DEPTNO;

Select * from Dep;
```

	DEPTNO	DEPTCOUNT
1	10	5
2	20	5
3	30	5

Q7. For a table that is not in BCNF, there is a certain similarity with a table that is in 2NF and not 3NF. Explain this parallel, and also explain the consequences in terms of RR1 and RR2. Show an example, with functional dependency diagrams.

S7. A table is in BCNF if it is in 3NF and iff every determinant is a candidate key. A table is in 3NF if there's a transitive dependence. The transitive dependency occurs if a non-prime attribute is dependent on another non-prime attribute, yet in BCNF in every X→Y dependence, the X must be a super key. This dependency is rather a stricter version of 3NF, and it is rare to find a relation that is in 3NF without being in BCNF.

RR1 says a table must be about only one entity. In this table, non-prime Company is dependent on the other non-prime attribute Company Phone. Thus, 2NF and RR1 are consistent. 3NF improves compliances with Robert's rule and BCNF and RR2 are consistent.



ItemID	Items	Company	Company Phone	Price
I001	iPhone 12	Apple Inc.	1 800-275-2273	799
I002	Apple TV	Apple Inc.	1 800-275-2273	199
I003	Macbook Pro 16	Apple Inc.	1 800-275-2273	2399
I004	Samsung S21	Samsung Inc.	1-888-699-6067	1399
I005	Play Station 5	Sony Computer Entertainment	1 800-538-7550	699

Q8. You have graduated and received an A in CSCI 6442. Congratulations!
You are hired by GWU as their Enterprise Data Architect. You are asked to advise on the data

model for a system that admits people to GWU buildings. Some of these people are GWU employees, some are GWU students, some work for contractors to GWU and some have no connection with GWU at all and may simply be visiting the university or hospital. Some are very low-paid employees of contractors, such as janitors, who may not have cell phones. The visitors may be U.S. citizens or citizens of other countries. They may be visiting for a long period, or they may be staying locally for only a short time, such as presenting a paper to a department colloquium.

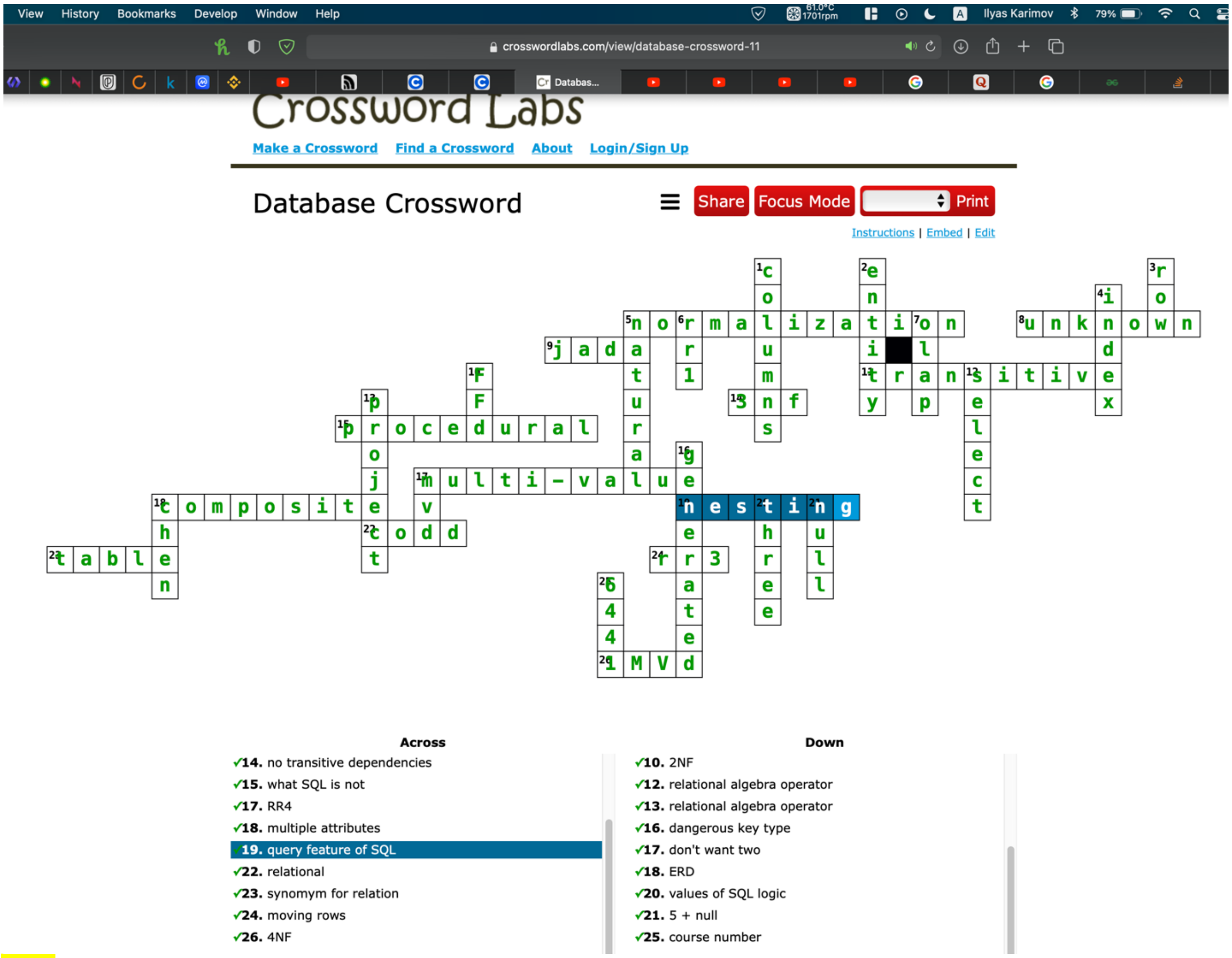
The way the system works is that a person who wants to have access to appropriate buildings first enters their personal information into the GWU visitor control database, including name, address, telephone number, citizenship, SSN (if one has been assigned), student number (if a GWU student), employee number (if a GWU employee) and passport number if available. A card is issued that the person uses to gain admission to campus buildings. The person scans the card outside a door to gain admission.

For the permitted visitor table, what will you use as key? Why? Will it be a generated key or a natural key? Why?

S8. The attributes we were introduced could have been used as primary keys, however none of the primary keys must contain NULL constraint. The citizen of other countries will not own SSN, the person who is outside of GWU will not own a GWID, the passport might expire and its number changes (this is the case in my country), not every person may have a telephone number, or the person may have no address, or two different people may be from the same family living in the same house. That’s why I would definitely use a generated key to be prepared for NULL values and have no issues.

Q9. Click on the link below, which will take you to a website with a database crossword puzzle. When you have finished the puzzle, copy the puzzle, with your answers, and paste it below the link:

S9.


View History Bookmarks Develop Window Help
crosswordlabs.com/view/database-crossword-11
Crossword Labs
Make a Crossword
Find a Crossword
About
Login/Sign Up
Share
Focus Mode
Print
Instructions | Embed | Edit

Database Crossword

Across

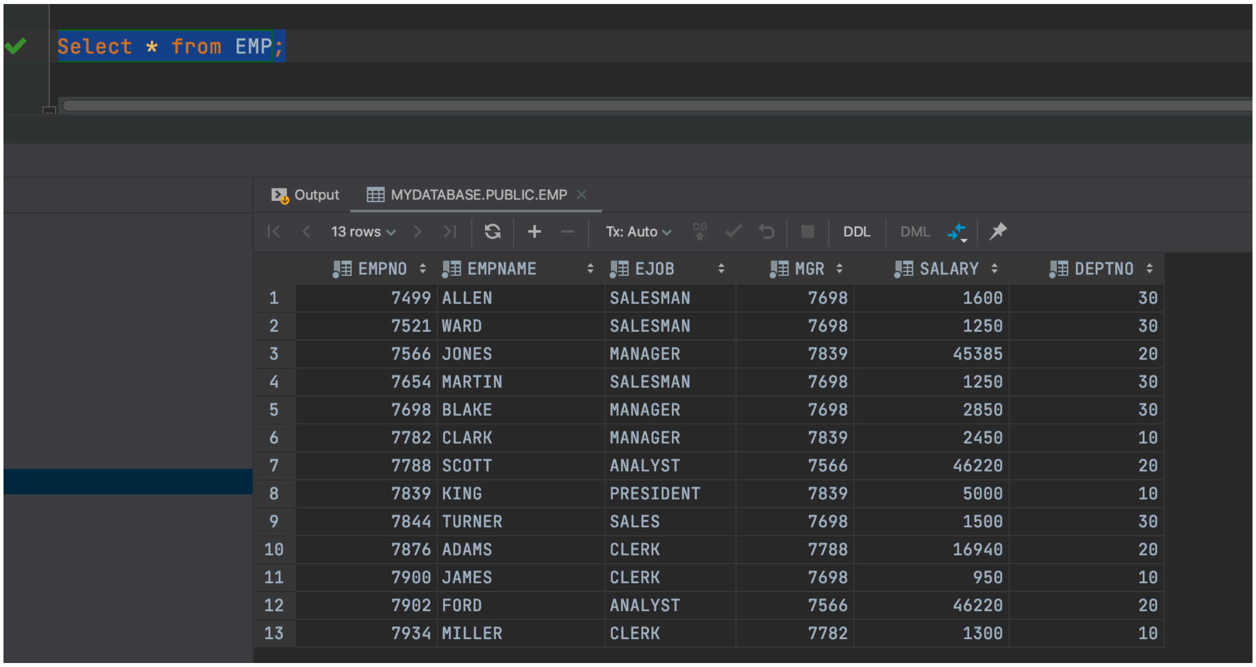
✓**14.** no transitive dependencies
✓**15.** what SQL is not
✓**17.** RR4
✓**18.** multiple attributes
19. query feature of SQL
✓**22.** relational
✓**23.** synomym for relation
✓**24.** moving rows
✓**26.** 4NF

Down

✓**10.** 2NF
✓**12.** relational algebra operator
✓**13.** relational algebra operator
✓**16.** dangerous key type
✓**17.** don't want two
✓**18.** ERD
✓**20.** values of SQL logic
✓**21.** 5 + null
✓**25.** course number

Q10. For the employee table, write a single UPDATE statement to lower the salary of any employee who earns more than their supervisor to 10% less than their supervisor&apos;s salary. Demonstrate that it works.

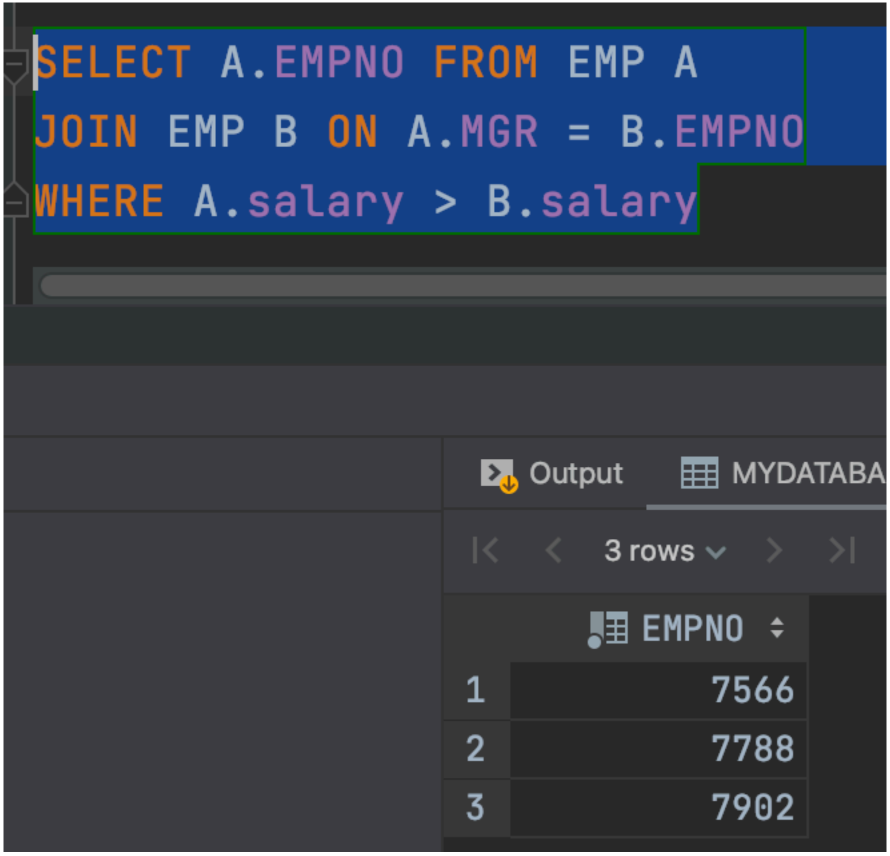
S10. The table is as below.



```sql
Select * from EMP;
```

	EMPNO	EMPNAME	EJOB	MGR	SALARY	DEPTNO
1	7499	ALLEN	SALESMAN	7698	1600	30
2	7521	WARD	SALESMAN	7698	1250	30
3	7566	JONES	MANAGER	7839	45385	20
4	7654	MARTIN	SALESMAN	7698	1250	30
5	7698	BLAKE	MANAGER	7698	2850	30
6	7782	CLARK	MANAGER	7839	2450	10
7	7788	SCOTT	ANALYST	7566	46220	20
8	7839	KING	PRESIDENT	7839	5000	10
9	7844	TURNER	SALES	7698	1500	30
10	7876	ADAMS	CLERK	7788	16940	20
11	7900	JAMES	CLERK	7698	950	10
12	7902	FORD	ANALYST	7566	46220	20
13	7934	MILLER	CLERK	7782	1300	10

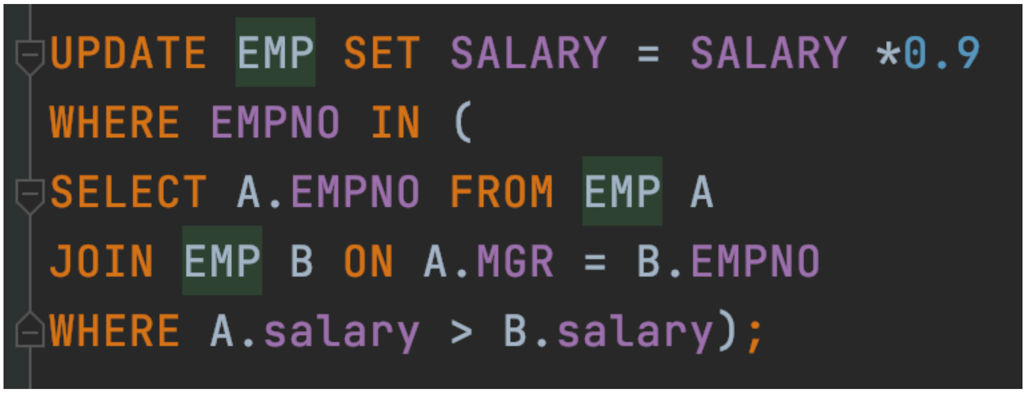
Firstly, finding those EMPNOs whose salary is higher than their managers.



```sql
SELECT A.EMPNO FROM EMP A
JOIN EMP B ON A.MGR = B.EMPNO
WHERE A.salary > B.salary
```

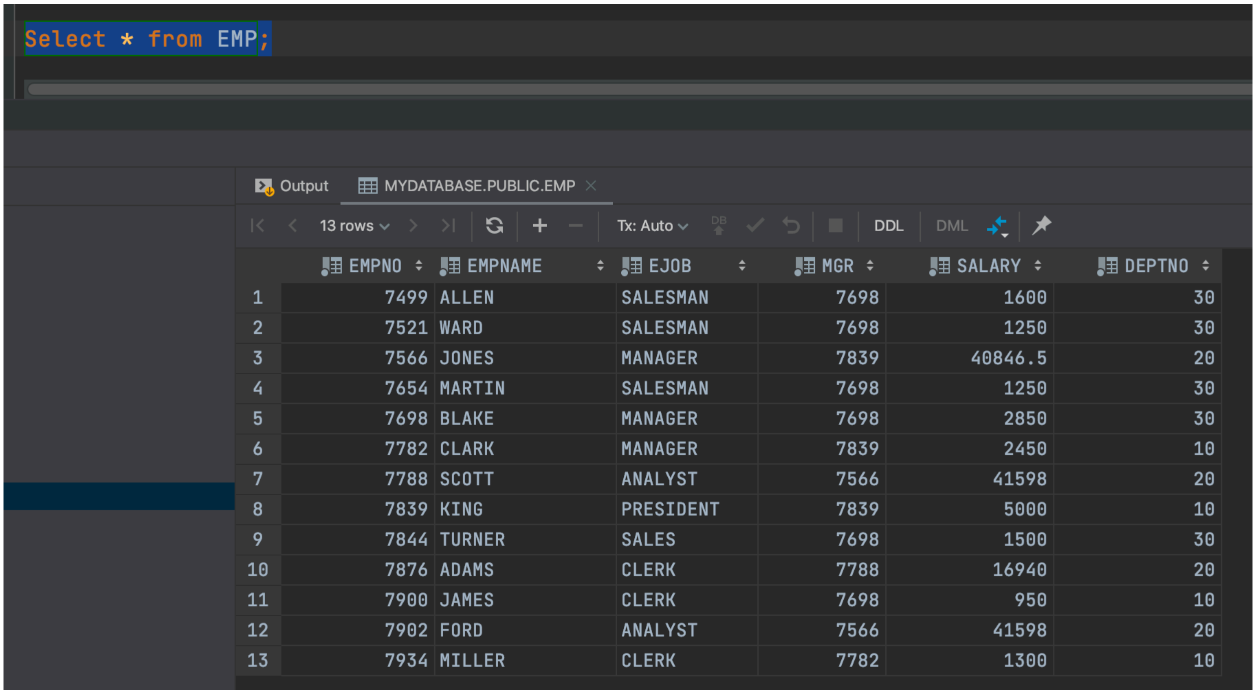
	EMPNO
1	7566
2	7788
3	7902

Then lowering the salary 10%.



```sql
UPDATE EMP SET SALARY = SALARY *0.9
WHERE EMPNO IN (
SELECT A.EMPNO FROM EMP A
JOIN EMP B ON A.MGR = B.EMPNO
WHERE A.salary > B.salary);
```

Salaries of EMPNOs 7566, 7788, 7902 have been lowered.



```sql
Select * from EMP;
```

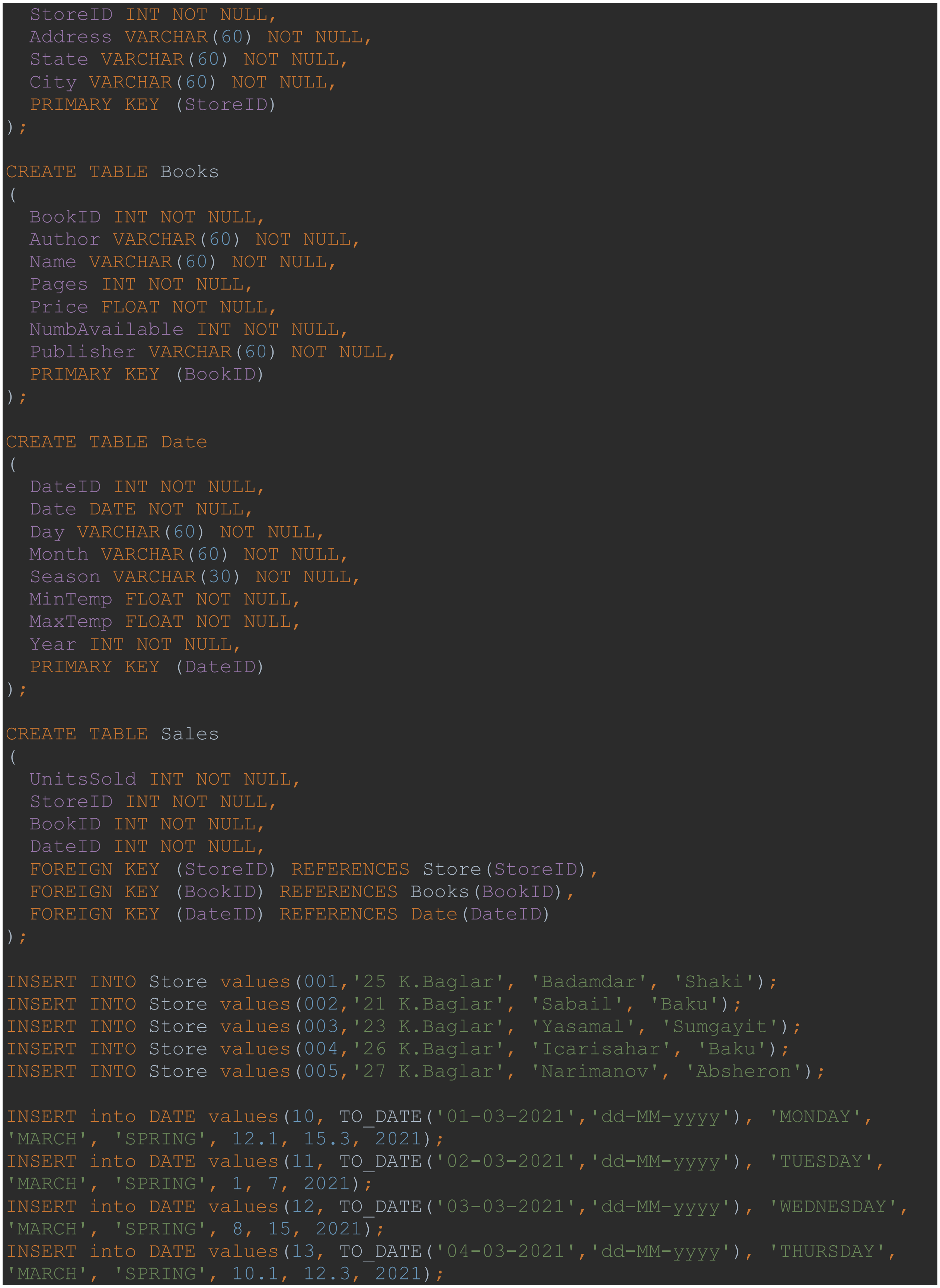
	EMPNO	EMPNAME	EJOB	MGR	SALARY	DEPTNO
1	7499	ALLEN	SALESMAN	7698	1600	30
2	7521	WARD	SALESMAN	7698	1250	30
3	7566	JONES	MANAGER	7839	40846.5	20
4	7654	MARTIN	SALESMAN	7698	1250	30
5	7698	BLAKE	MANAGER	7698	2850	30
6	7782	CLARK	MANAGER	7839	2450	10
7	7788	SCOTT	ANALYST	7566	41598	20
8	7839	KING	PRESIDENT	7839	5000	10
9	7844	TURNER	SALES	7698	1500	30
10	7876	ADAMS	CLERK	7788	16940	20
11	7900	JAMES	CLERK	7698	950	10
12	7902	FORD	ANALYST	7566	41598	20
13	7934	MILLER	CLERK	7782	1300	10

EXTRA

1. What single thing do you like most about the course?
   I like the professor's attitude in this course. He behaves so differently that I do not experience the same amount stress as I do in other courses and fulfill my responsibilities (tasks) on time, hence focus on learning more. I also like how he respects and explains to us his own way when we share our ideas with him, even if it's ridiculous.
2. What single thing do you like least about the course?
   It would be a great upgrade to receive individual feedback on our homework, so that we will be knowing whether what we have contemplated and written is correct or we should work on it to fix our mistakes.

APPENDICES (**SQL Queries**)
Appendix 3.

```sql
 CREATE TABLE Store
(
```

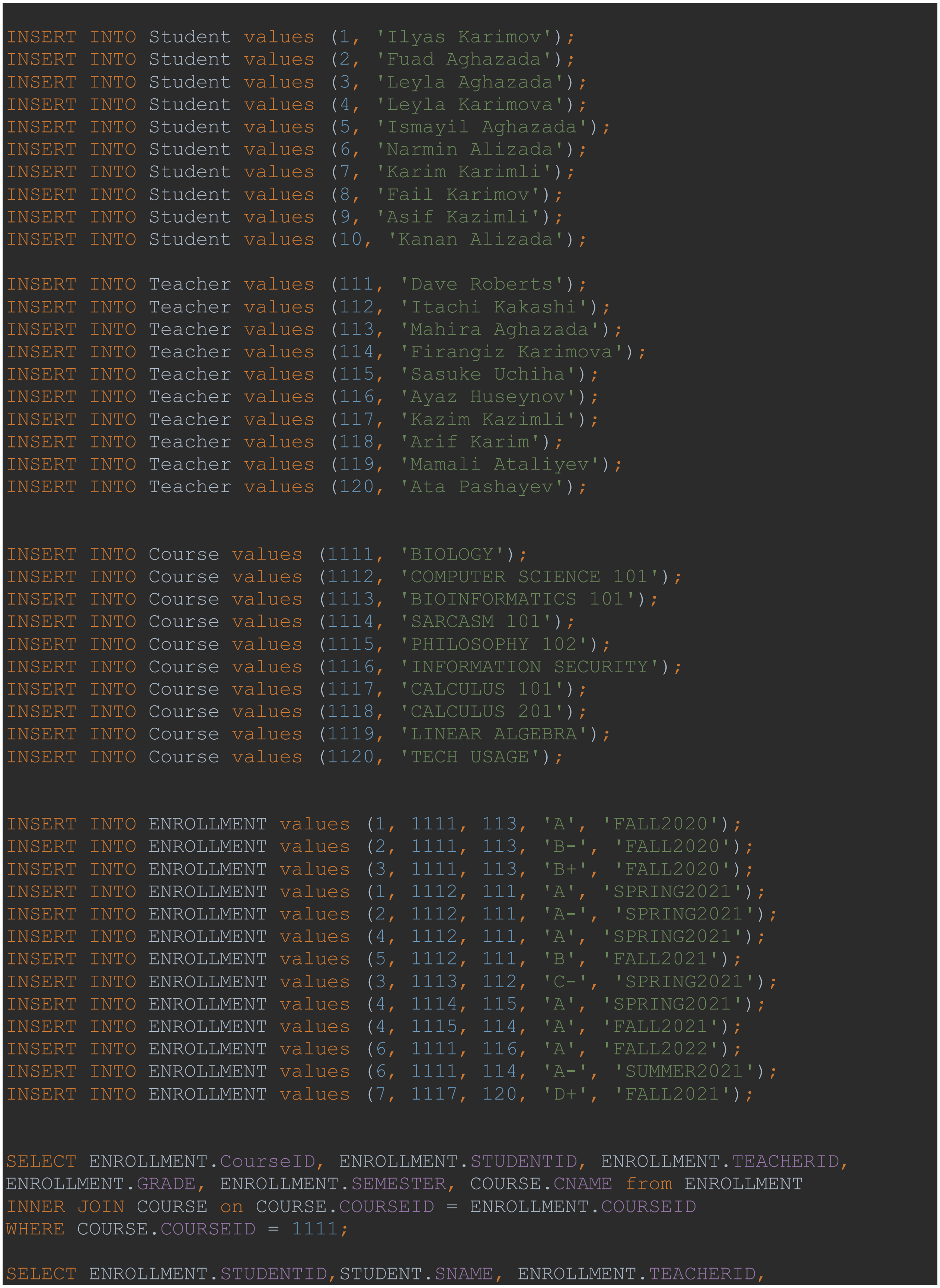


```
  StoreID INT NOT NULL,
  Address VARCHAR(60) NOT NULL,
  State VARCHAR(60) NOT NULL,
  City VARCHAR(60) NOT NULL,
  PRIMARY KEY (StoreID)
);

CREATE TABLE Books
(
  BookID INT NOT NULL,
  Author VARCHAR(60) NOT NULL,
  Name VARCHAR(60) NOT NULL,
  Pages INT NOT NULL,
  Price FLOAT NOT NULL,
  NumbAvailable INT NOT NULL,
  Publisher VARCHAR(60) NOT NULL,
  PRIMARY KEY (BookID)
);

CREATE TABLE Date
(
  DateID INT NOT NULL,
  Date DATE NOT NULL,
  Day VARCHAR(60) NOT NULL,
  Month VARCHAR(60) NOT NULL,
  Season VARCHAR(30) NOT NULL,
  MinTemp FLOAT NOT NULL,
  MaxTemp FLOAT NOT NULL,
  Year INT NOT NULL,
  PRIMARY KEY (DateID)
);

CREATE TABLE Sales
(
  UnitsSold INT NOT NULL,
  StoreID INT NOT NULL,
  BookID INT NOT NULL,
  DateID INT NOT NULL,
  FOREIGN KEY (StoreID) REFERENCES Store(StoreID),
  FOREIGN KEY (BookID) REFERENCES Books(BookID),
  FOREIGN KEY (DateID) REFERENCES Date(DateID)
);

INSERT INTO Store values(001,'25 K.Baglar', 'Badamdar', 'Shaki');
INSERT INTO Store values(002,'21 K.Baglar', 'Sabail', 'Baku');
INSERT INTO Store values(003,'23 K.Baglar', 'Yasamal', 'Sumgayit');
INSERT INTO Store values(004,'26 K.Baglar', 'Icarisahar', 'Baku');
INSERT INTO Store values(005,'27 K.Baglar', 'Narimanov', 'Absheron');

INSERT into DATE values(10, TO_DATE('01-03-2021','dd-MM-yyyy'), 'MONDAY',
'MARCH', 'SPRING', 12.1, 15.3, 2021);
INSERT into DATE values(11, TO_DATE('02-03-2021','dd-MM-yyyy'), 'TUESDAY',
'MARCH', 'SPRING', 1, 7, 2021);
INSERT into DATE values(12, TO_DATE('03-03-2021','dd-MM-yyyy'), 'WEDNESDAY',
'MARCH', 'SPRING', 8, 15, 2021);
INSERT into DATE values(13, TO_DATE('04-03-2021','dd-MM-yyyy'), 'THURSDAY',
'MARCH', 'SPRING', 10.1, 12.3, 2021);
```

```
INSERT into DATE values(14, TO_DATE('05-06-2021','dd-MM-yyyy'), 'SATURDAY',
'JUNE', 'SUMMER', 28.1, 33.3, 2021);

INSERT INTO BOOKS values(100, 'Dave Roberts', 'Database Managements',  1324,
243.1, 5,'Pearl');
INSERT INTO BOOKS values(101, 'Ilyas Karimov','Depression and Anxiety',
2552, 300, 2,'Pearl'  );
INSERT INTO BOOKS values(102, 'Ilyas Karimov', 'Sarcasms for dummies', 1578,
133, 52,'Limax' );
INSERT INTO BOOKS values(103, 'Fuad Aghazada','How to master programming
skills',1556, 213.1, 51, 'Kstamonian');

INSERT INTO Sales values (1, 001, 101, 10);
INSERT INTO Sales values (1, 002, 101, 12);
INSERT INTO Sales values (5, 003, 102, 10);
INSERT INTO Sales values (2, 5, 100, 14);
INSERT INTO Sales values (12, 4, 103, 11);

Select * from books;
Select * from Store;
Select * from SALES;

Select BOOKS.Author, BOOKS.NAME, Sales.DateID, SALES.STOREID from SALES
INNER JOIN BOOKS ON BOOKS.BookID = SALES.BOOKID;
```

Appendix 5.

```
CREATE TABLE Student
(
  StudentID INT NOT NULL,
  SName VARCHAR(60) NOT NULL,
  PRIMARY KEY (StudentID)
);

CREATE TABLE Teacher
(
  TeacherID INT NOT NULL,
  TName VARCHAR(60) NOT NULL,
  PRIMARY KEY (TeacherID)
);

CREATE TABLE Course
(
  CourseID INT NOT NULL,
  CName VARCHAR(60) NOT NULL,
  PRIMARY KEY (CourseID)
);

CREATE TABLE Enrollment
(
  StudentID INT NOT NULL,
  CourseID INT NOT NULL,
  TeacherID INT NOT NULL,
  Grade VARCHAR(60) NOT NULL,
  Semester VARCHAR(60) NOT NULL
);
```



```
INSERT INTO Student values (1, 'Ilyas Karimov');
INSERT INTO Student values (2, 'Fuad Aghazada');
INSERT INTO Student values (3, 'Leyla Aghazada');
INSERT INTO Student values (4, 'Leyla Karimova');
INSERT INTO Student values (5, 'Ismayil Aghazada');
INSERT INTO Student values (6, 'Narmin Alizada');
INSERT INTO Student values (7, 'Karim Karimli');
INSERT INTO Student values (8, 'Fail Karimov');
INSERT INTO Student values (9, 'Asif Kazimli');
INSERT INTO Student values (10, 'Kanan Alizada');

INSERT INTO Teacher values (111, 'Dave Roberts');
INSERT INTO Teacher values (112, 'Itachi Kakashi');
INSERT INTO Teacher values (113, 'Mahira Aghazada');
INSERT INTO Teacher values (114, 'Firangiz Karimova');
INSERT INTO Teacher values (115, 'Sasuke Uchiha');
INSERT INTO Teacher values (116, 'Ayaz Huseynov');
INSERT INTO Teacher values (117, 'Kazim Kazimli');
INSERT INTO Teacher values (118, 'Arif Karim');
INSERT INTO Teacher values (119, 'Mamali Ataliyev');
INSERT INTO Teacher values (120, 'Ata Pashayev');


INSERT INTO Course values (1111, 'BIOLOGY');
INSERT INTO Course values (1112, 'COMPUTER SCIENCE 101');
INSERT INTO Course values (1113, 'BIOINFORMATICS 101');
INSERT INTO Course values (1114, 'SARCASM 101');
INSERT INTO Course values (1115, 'PHILOSOPHY 102');
INSERT INTO Course values (1116, 'INFORMATION SECURITY');
INSERT INTO Course values (1117, 'CALCULUS 101');
INSERT INTO Course values (1118, 'CALCULUS 201');
INSERT INTO Course values (1119, 'LINEAR ALGEBRA');
INSERT INTO Course values (1120, 'TECH USAGE');


INSERT INTO ENROLLMENT values (1, 1111, 113, 'A', 'FALL2020');
INSERT INTO ENROLLMENT values (2, 1111, 113, 'B-', 'FALL2020');
INSERT INTO ENROLLMENT values (3, 1111, 113, 'B+', 'FALL2020');
INSERT INTO ENROLLMENT values (1, 1112, 111, 'A', 'SPRING2021');
INSERT INTO ENROLLMENT values (2, 1112, 111, 'A-', 'SPRING2021');
INSERT INTO ENROLLMENT values (4, 1112, 111, 'A', 'SPRING2021');
INSERT INTO ENROLLMENT values (5, 1112, 111, 'B', 'FALL2021');
INSERT INTO ENROLLMENT values (3, 1113, 112, 'C-', 'SPRING2021');
INSERT INTO ENROLLMENT values (4, 1114, 115, 'A', 'SPRING2021');
INSERT INTO ENROLLMENT values (4, 1115, 114, 'A', 'FALL2021');
INSERT INTO ENROLLMENT values (6, 1111, 116, 'A', 'FALL2022');
INSERT INTO ENROLLMENT values (6, 1111, 114, 'A-', 'SUMMER2021');
INSERT INTO ENROLLMENT values (7, 1117, 120, 'D+', 'FALL2021');


SELECT ENROLLMENT.CourseID, ENROLLMENT.STUDENTID, ENROLLMENT.TEACHERID,
ENROLLMENT.GRADE, ENROLLMENT.SEMESTER, COURSE.CNAME from ENROLLMENT
INNER JOIN COURSE on COURSE.COURSEID = ENROLLMENT.COURSEID
WHERE COURSE.COURSEID = 1111;

SELECT ENROLLMENT.STUDENTID,STUDENT.SNAME, ENROLLMENT.TEACHERID,
ENROLLMENT.GRADE, ENROLLMENT.SEMESTER from ENROLLMENT
INNER JOIN STUDENT on STUDENT.StudentID = ENROLLMENT.StudentID
WHERE STUDENT.StudentID = 1;
```

Appendix 6.

```
CREATE TABLE DEPARTMENT(
    DEPTNO INT NOT NULL,
    DEPTNAME VARCHAR(60)
);

CREATE TABLE EMPLOYEE(
    EMPNO INT NOT NULL,
    EMPNAME VARCHAR(30) NOT NULL,
    JOBNAME VARCHAR(30) NOT NULL,
    SAL INT NOT NULL,
    DEPTNO INT NOT NULL,
    PRIMARY KEY (EMPNO),
    FOREIGN KEY (DEPTNO) REFERENCES DEPARTMENT(DEPTNO)
);

INSERT INTO DEPARTMENT VALUES (10, 'Education');
INSERT INTO DEPARTMENT VALUES (20, 'Marketing');
INSERT INTO DEPARTMENT VALUES (30, 'Sales');

INSERT INTO EMPLOYEE VALUES (100, 'Ilyas Karimov', 'Head of
Education', 1200, 10);
INSERT INTO EMPLOYEE VALUES (101, 'Aytac Nuraddinova', 'Teaching
Supervisor', 600, 10);
INSERT INTO EMPLOYEE VALUES (102, 'Narmin Mirzayeva', 'Education
Intern', 200, 10);
INSERT INTO EMPLOYEE VALUES (103, 'Rustam Alizada', 'Programming
Teacher', 350, 10);
INSERT INTO EMPLOYEE VALUES (104, 'Elnara Nabiyeva', 'CEO',
1800, 10);

INSERT INTO EMPLOYEE VALUES (105, 'Leyla Miriyeva', 'Head of
Marketing', 1500, 20);
INSERT INTO EMPLOYEE VALUES (106, 'Khadija Salimova',
'Copywriter', 500, 20);
INSERT INTO EMPLOYEE VALUES (107, 'Nigar Babayeva', 'Designer',
1000, 20);
INSERT INTO EMPLOYEE VALUES (108, 'Shahin Balayev', 'Designer',
800, 20);
INSERT INTO EMPLOYEE VALUES (109, 'Ali Karamzada', 'Design
Intern', 200, 20);
```

```
INSERT INTO EMPLOYEE VALUES (110, 'Sona Orucova', 'Head of
Sales', 500, 30);
INSERT INTO EMPLOYEE VALUES (111, 'Nazrin Tagiyeva', 'Sales
Coordinator', 300, 30);
INSERT INTO EMPLOYEE VALUES (112, 'Rauf Atakiwiyev', 'Sales
Intern ', 150, 30);
INSERT INTO EMPLOYEE VALUES (113, 'Nadejda Ayxanova', 'Sales
Supervisor ', 150, 30);
INSERT INTO EMPLOYEE VALUES (114, 'Teymur Shukurov', 'Project
Manager ', 1420, 30);

create view DEPT20 as Select EMPNO, EMPName from EMPLOYEE where
DEPTNO = 20;

select * from DEPT20;

UPDATE DEPT20
SET EMPNAME = 'Kanan Karamzada' where EMPNO = 109;

CREATE VIEW Dep(DEPTNO, DEPTCOUNT) AS
SELECT DEPTNO, COUNT(DEPTNO)
FROM EMPLOYEE
GROUP BY DEPTNO;

Select * from Dep;

SELECT * FROM DEPTS;


SELECT * from EMPLOYEE;
-- Create a department table like the ones discussed in class,
with departments 10, 20, and 30, each with 5 employees
```

Appendix 10.

```

CREATE TABLE EMP
(
  EMPNO INT NOT NULL,
  EMPNAME VARCHAR(60) NOT NULL,
  EJOB VARCHAR(60) NOT NULL,
  MGR INT NOT NULL,
  SALARY FLOAT NOT NULL,
  DEPTNO INT NOT NULL
);

```

```
INSERT INTO EMP VALUES (7369, 'SMITH', 'CLERK', 7902, 12323,
20);
INSERT INTO EMP VALUES (7499, 'ALLEN', 'SALESMAN', 7698, 1600,
30);
INSERT INTO EMP VALUES (7521, 'WARD', 'SALESMAN', 7698, 1250,
30);
INSERT INTO EMP VALUES (7566, 'JONES', 'MANAGER', 7839, 45385
,20);
INSERT INTO EMP VALUES (7654, 'MARTIN', 'SALESMAN', 7698, 1250,
30);
INSERT INTO EMP VALUES (7698, 'BLAKE', 'MANAGER', 7698,2850
,30);
INSERT INTO EMP VALUES (7782, 'CLARK', 'MANAGER', 7839,2450
,10);
INSERT INTO EMP VALUES (7788, 'SCOTT', 'ANALYST', 7566, 46220
,20);
INSERT INTO EMP VALUES (7839, 'KING', 'PRESIDENT', 7839,5000
,10);
INSERT INTO EMP VALUES (7844, 'TURNER', 'SALES', 7698, 1500
,30);
INSERT INTO EMP VALUES (7876, 'ADAMS', 'CLERK', 7788,16940 ,20);
INSERT INTO EMP VALUES (7900, 'JAMES', 'CLERK', 7698, 950 ,10);
INSERT INTO EMP VALUES (7902, 'FORD', 'ANALYST', 7566 ,46220
,20);
INSERT INTO EMP VALUES (7934, 'MILLER', 'CLERK', 7782,1300 ,10);


Select * from EMP;

SELECT A.EMPNO FROM EMP A
JOIN EMP B ON A.MGR = B.EMPNO
WHERE A.salary > B.salary;

UPDATE EMP SET SALARY = SALARY *0.9
WHERE EMPNO IN (
SELECT A.EMPNO FROM EMP A
JOIN EMP B ON A.MGR = B.EMPNO
WHERE A.salary > B.salary);
```