CSI2132 Tutorial 5

The Structured Query Language (SQL)

Presented By: Rana Khalil

About Me

- Rana Khalil
- Master in Computer Science – 2nd year
- Email: rkhal101@uottawa.ca
- Office Hours: Tuesdays 12:00 PM – 1:00 PM in SITE 5000G
- Teach the next 3 tutorials

Chapter 4: “Basic SQL”

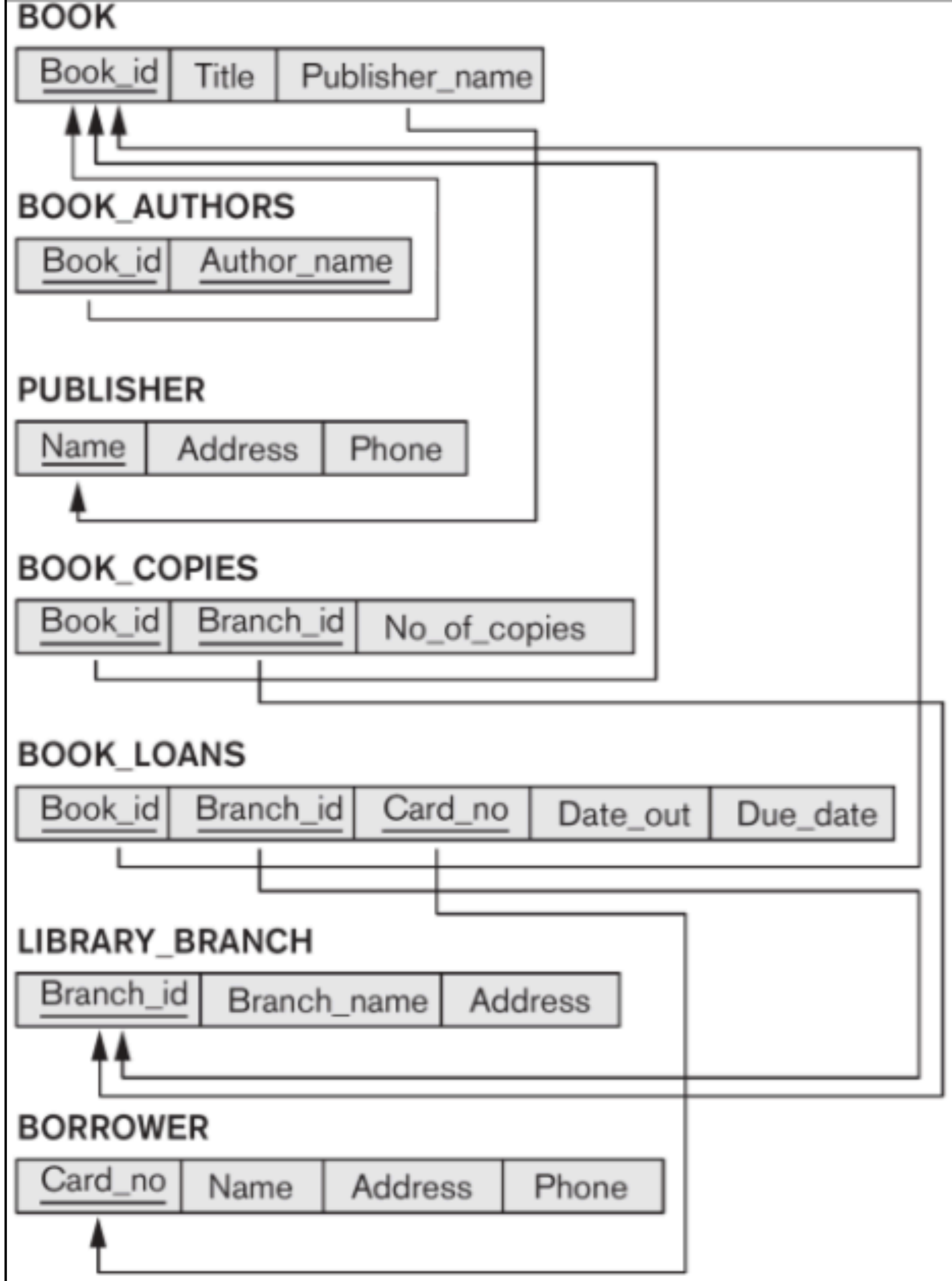
Exercise 4.7

Consider the schema for the LIBRARY database in Fig. 4.6

Choose the appropriate action (reject, cascade, set to null, set to default) for each referential integrity constraint, both for a deletion of a referenced tuple and for the update of a primary key attribute value in a referenced tuple. Justify your choices.

Actions on DELETE and UPDATE

- **CASCADE**: Make the same deletion/update in the referring tuple.
- **SET NULL**: Set the corresponding value in the referring tuple to null.
- **REJECT**: Don’t allow the deletion/update.



BOOK

Book_id	Title	Publisher_name

BOOK_AUTHORS

Book_id	Author_name

PUBLISHER

Name	Address	Phone

BOOK_COPIES

Book_id	Branch_id	No_of_copies

BOOK_LOANS

Book_id	Branch_id	Card_no	Date_out	Due_date

LIBRARY_BRANCH

Branch_id	Branch_name	Address

BORROWER

Card_no	Name	Address	Phone

Figure 4.6
A relational database schema for a LIBRARY database.

Chapter 4: “Basic SQL”

BOOK_AUTHORS.(BookId) --> BOOK.(BookId)

ON DELETE CASCADE
ON UPDATE CASCADE

Automatically propagate the deletion or change of a BOOK to the referencing BOOK_AUTHORS.

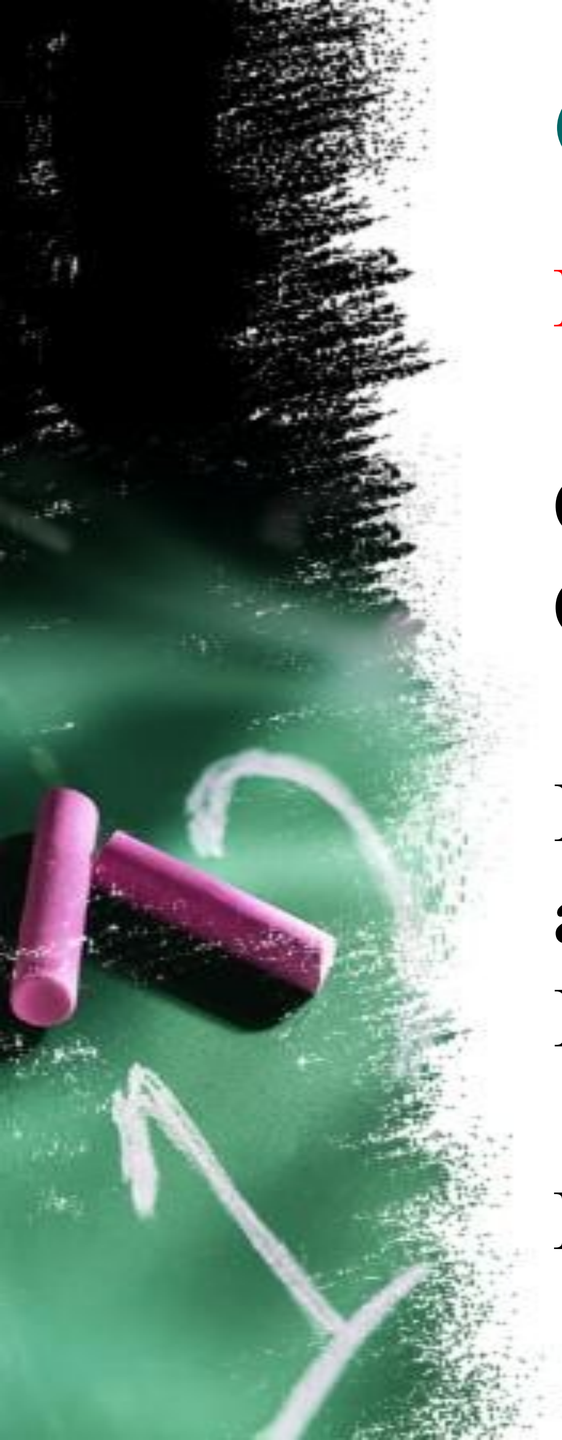
Chapter 4: “Basic SQL”

BOOK.(PublisherName) --> PUBLISHER.(Name)

ON DELETE REJECT
ON UPDATE CASCADE

Do not delete a PUBLISHER tuple which has linked BOOK tuples.

Update the PUBLISHER’s name on all BOOK tuples which refer to it.



Chapter 4: “Basic SQL”

BOOK_LOANS.(BookID) --> BOOK.(BookID)

ON DELETE CASCADE
ON UPDATE CASCADE

If a BOOK record is deleted, then delete all its associated BOOK_LOAN records.
Idem with updates.

REJECT on DELETE also possible!

Chapter 4: “Basic SQL”

BOOK_COPIES.(BookID) --> BOOK.(BookID)

ON DELETE CASCADE
ON UPDATE CASCADE

If a BOOK record is deleted, then delete all its associated BOOK_COPIES tuples.
Do likewise with updates.

Chapter 4: “Basic SQL”

BOOK_LOANS.(CardNo) --> BORROWER.(CardNo)

ON DELETE CASCADE
ON UPDATE CASCADE

If a BORROWER record is deleted, then delete all its associated BOOK_LOANS tuples.
Do likewise with updates.

REJECT on DELETE also possible!

Chapter 4: “Basic SQL”

BOOK_COPIES.(BranchID) -->
LIBRARY_BRANCH.(BranchID)

ON DELETE CASCADE
ON UPDATE CASCADE

If a LIBRARY_BRANCH record is deleted, then delete all its linked BOOK_COPIES tuples.
Do likewise with updates.

REJECT on DELETE also possible!

Chapter 4: “Basic SQL”

BOOK_LOANS.(BranchID) -->
LIBRARY_BRANCH.(BranchID)

ON DELETE CASCADE
ON UPDATE CASCADE

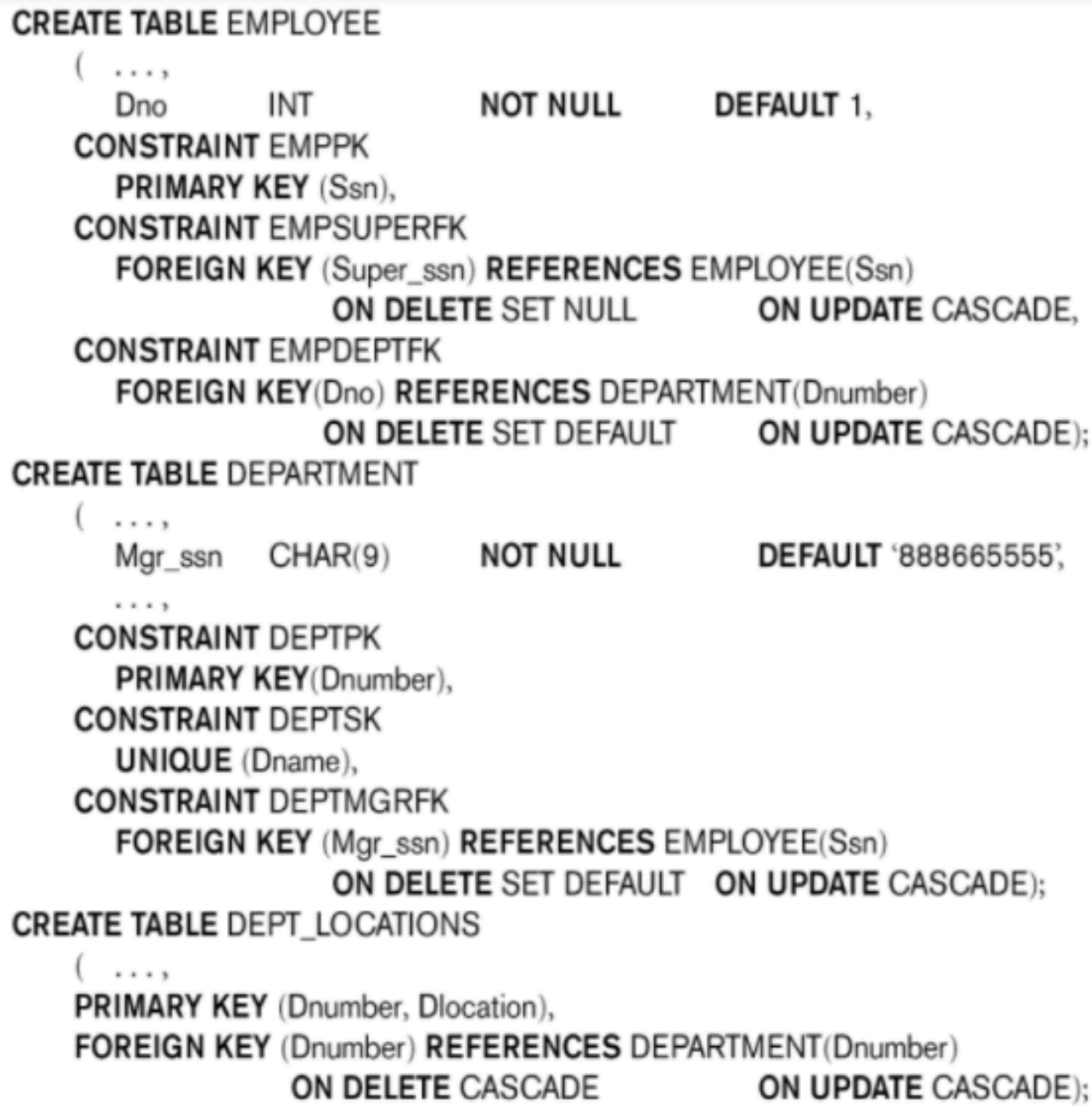
If a LIBRARY_BRANCH record is deleted, then delete all its linked BOOK_LOANS tuples.
Do likewise with updates.

REJECT on DELETE also possible!

Chapter 4: “Basic SQL”

Exercise 4.15

Consider the EMPLOYEE table’s constraint EMPSUPERFK as in Fig 4.2



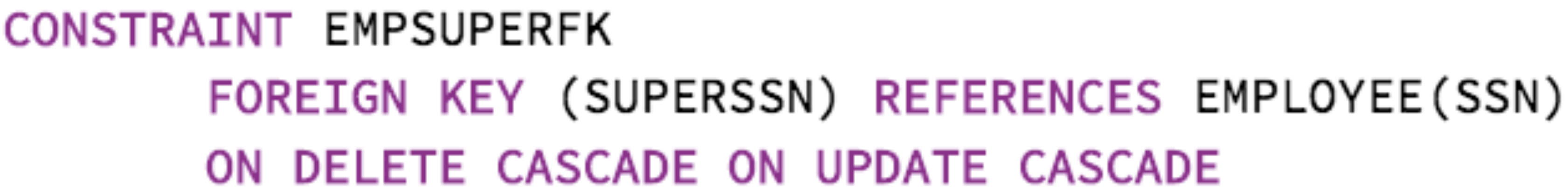
```
CREATE TABLE EMPLOYEE
    (   . . . ,
        Dno         INT             NOT NULL        DEFAULT 1,
    CONSTRAINT EMPPK
        PRIMARY KEY (Ssn),
    CONSTRAINT EMPSUPERFK
        FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn)
                        ON DELETE SET NULL          ON UPDATE CASCADE,
    CONSTRAINT EMPDEPTFK
        FOREIGN KEY(Dno) REFERENCES DEPARTMENT(Dnumber)
                        ON DELETE SET DEFAULT       ON UPDATE CASCADE);
CREATE TABLE DEPARTMENT
    (   . . . ,
        Mgr_ssn     CHAR(9)         NOT NULL            DEFAULT '888665555',
        . . . ,
    CONSTRAINT DEPTPK
        PRIMARY KEY(Dnumber),
    CONSTRAINT DEPTSK
        UNIQUE (Dname),
    CONSTRAINT DEPTMGRFK
        FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn)
                        ON DELETE SET DEFAULT   ON UPDATE CASCADE);
CREATE TABLE DEPT_LOCATIONS
    (   . . . ,
    PRIMARY KEY (Dnumber, Dlocation),
    FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber)
                    ON DELETE CASCADE               ON UPDATE CASCADE);
```

Figure 4.2
Example illustrating how default attribute values and referential integrity triggered actions are specified in SQL.

Chapter 4: “Basic SQL”

Exercise 4.15

If the constraint is changed to read as follows:

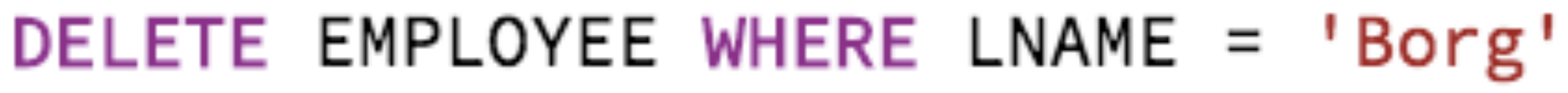


```
CONSTRAINT EMPSUPERFK
        FOREIGN KEY (SUPERSSN) REFERENCES EMPLOYEE(SSN)
        ON DELETE CASCADE ON UPDATE CASCADE
```

Chapter 4: “Basic SQL”

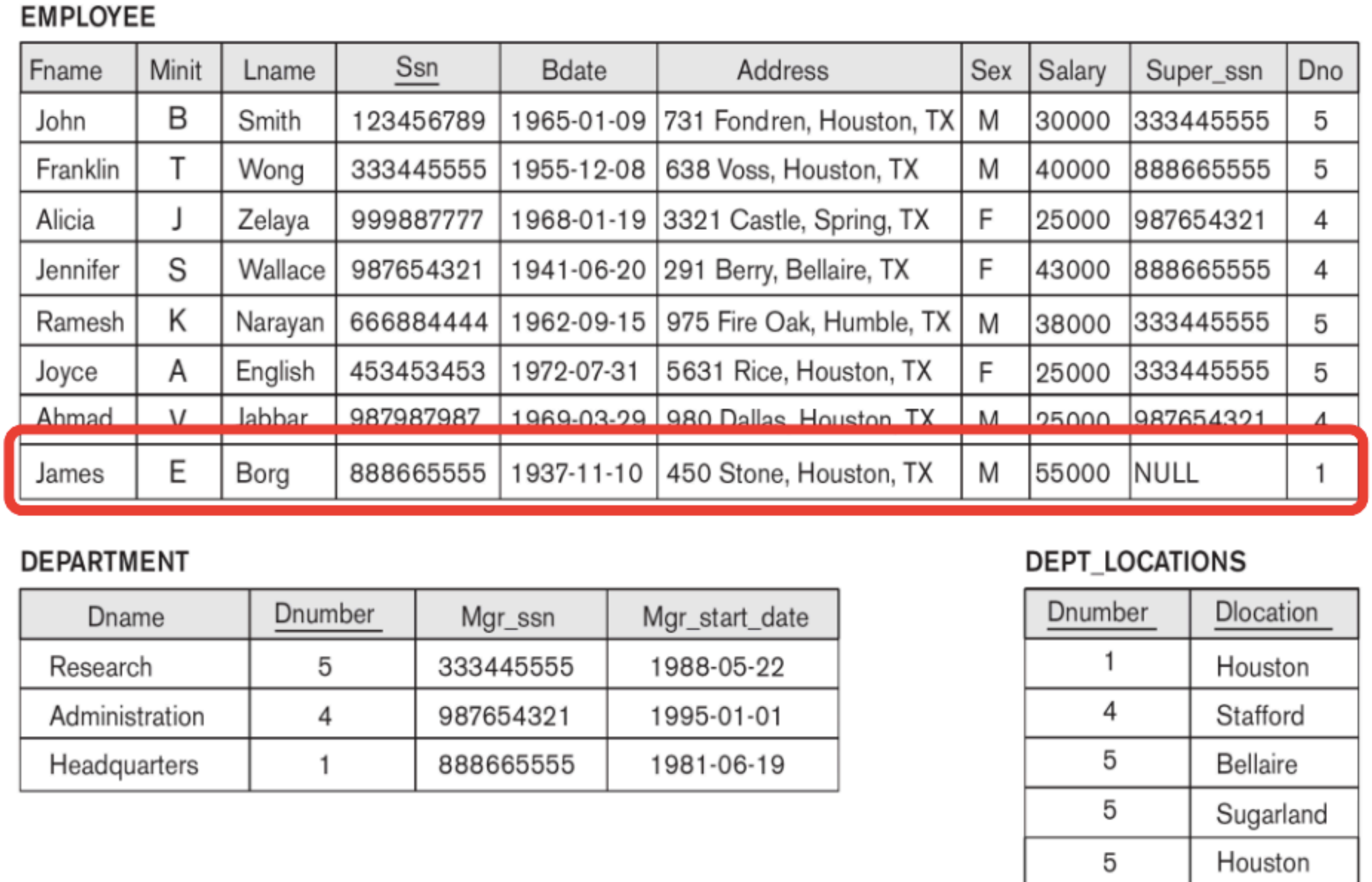
Exercise 4.15

What happens when the following command is run on the COMPANY database state shown in Fig. 3.6?



```
DELETE EMPLOYEE WHERE LNAME = 'Borg'
```

Figure 3.6
One possible database state for the COMPANY relational database schema.



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

Chapter 4: “Basic SQL”

Answer:

➢ Triggers a deletion of all subordinate records in James Borg’s supervision hierarchy.

Chapter 4: “Basic SQL”

Exercise 4.15

Is it better to CASCADE or SET NULL in case of EMPSUPERFK constraint ON DELETE?

Chapter 4: “Basic SQL”

Answer:

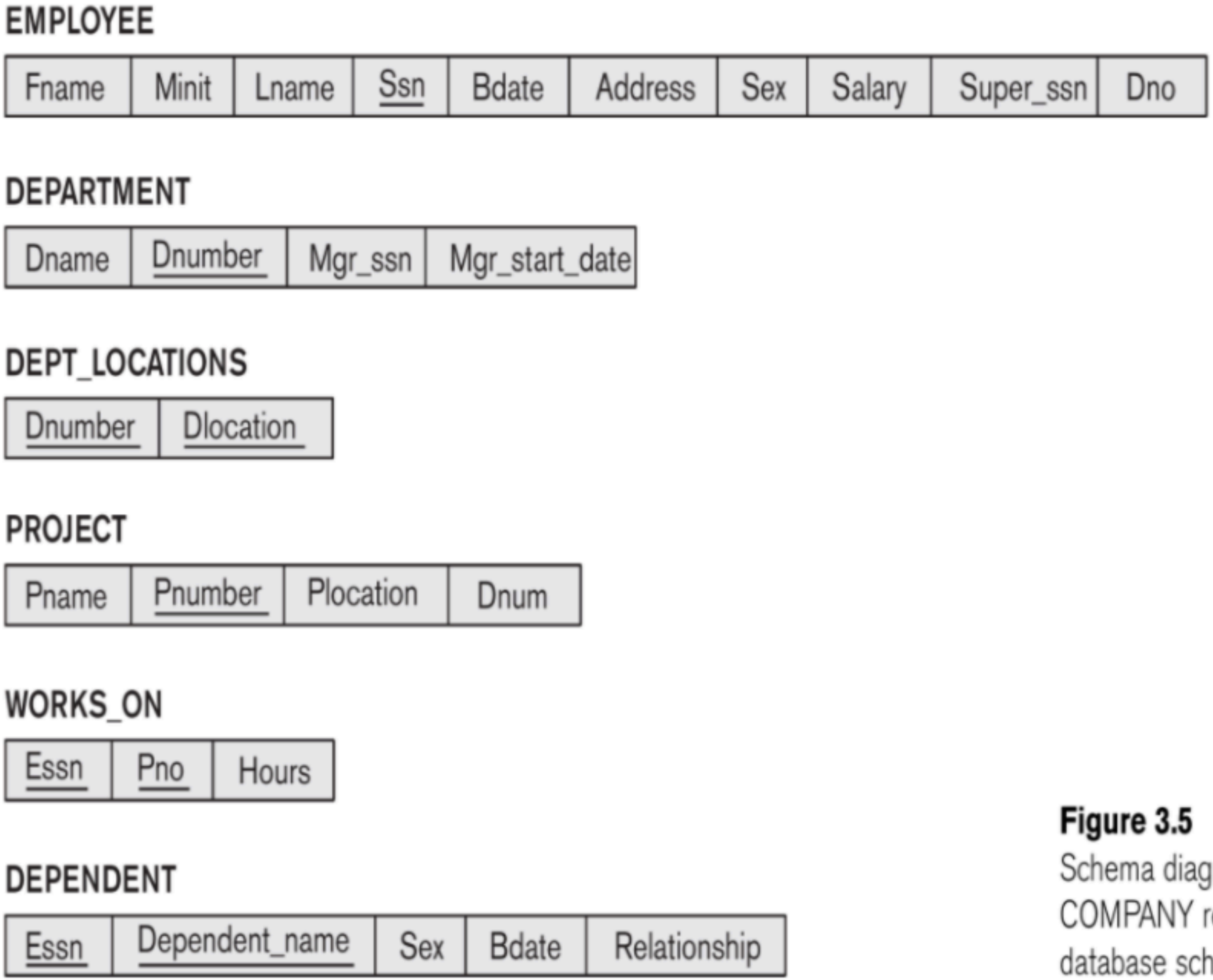
SET NULL is preferred, since an EMPLOYEE is not fired (deleted) when his/her supervisor is deleted.

Instead, the SUPERSSN field should be SET NULL so a new supervisor could be assigned later on.

Chapter 4: “More SQL”

Exercise 5.5

Specify the following additional SQL queries on the COMPANY database of Fig. 3.5



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Figure 3.5
Schema diagram for the COMPANY relational database schema.

Chapter 4: “More SQL”

Exercise 5.5

a) For each department whose average employee salary is over 30K, retrieve the department name and the number of employees working for it.

Chapter 4: “More SQL”

Exercise 5.5

a) For each department whose average employee salary is over 30K, retrieve the department name and the number of employees working for it.

```
SELECT
FROM
WHERE
```

Chapter 4: “More SQL”

Exercise 5.5

a) For each department whose average employee salary is over 30K, retrieve the department name and the number of employees working for it.

```
SELECT
FROM DEPARTMENT, EMPLOYEE
WHERE
```

Chapter 4: “More SQL”

Exercise 5.5

a) For each department whose average employee salary is over 30K, retrieve the department name and the number of employees working for it.

```
SELECT
FROM DEPARTMENT, EMPLOYEE
WHERE DNUMBER=DNO
```

Chapter 4: “More SQL”

Exercise 5.5

a) For each department whose average employee salary is over 30K, retrieve the department name and the number of employees working for it.

```
SELECT
FROM DEPARTMENT, EMPLOYEE
WHERE DNUMBER=DNO
GROUP BY DNAME
HAVING AVG (SALARY) > 30000
```

Chapter 4: “More SQL”

Exercise 5.5

a) For each department whose average employee salary is over 30K, retrieve the department name and the number of employees working for it.

```
SELECT DNAME, COUNT(*)
FROM DEPARTMENT, EMPLOYEE
WHERE DNUMBER=DNO
GROUP BY DNAME
HAVING AVG (SALARY) > 30000
```

Chapter 4: “More SQL”

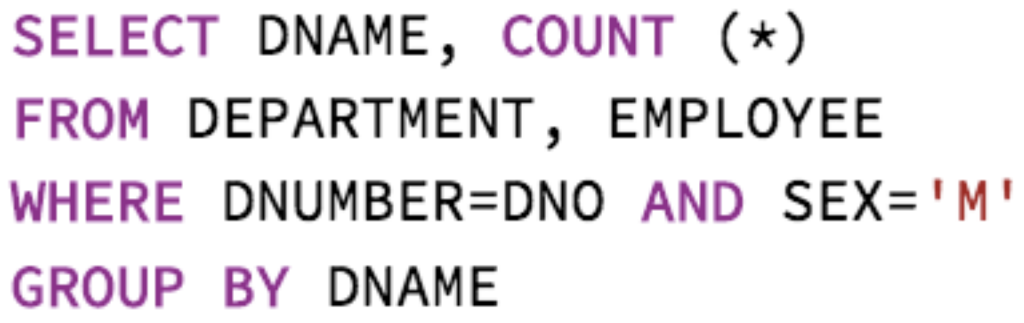
Exercise 5.5

b) Suppose we want the number of male employees in each department rather than all employees.
Can we specify this in SQL? Why or why not?

Yes, via a nested query

Chapter 4: “More SQL”

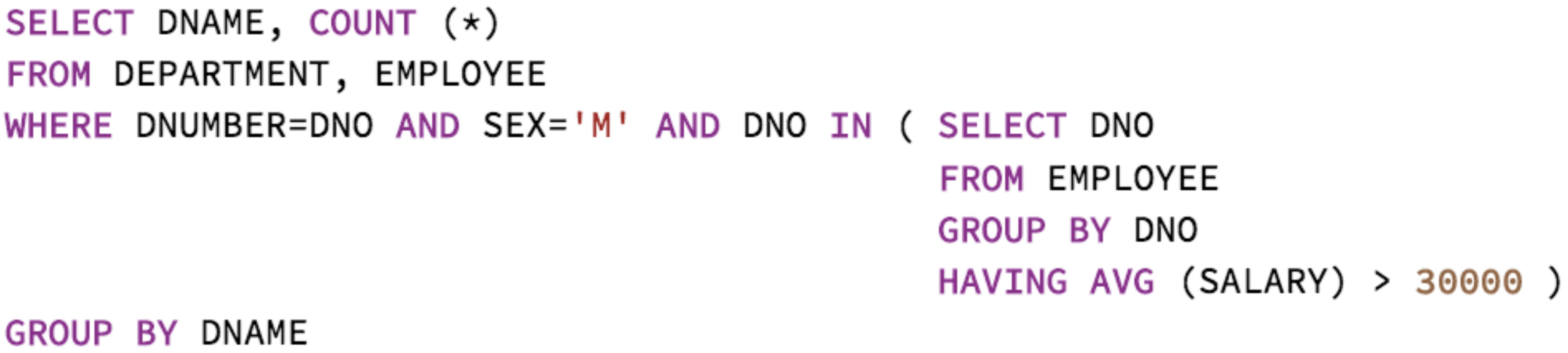
b) For each department whose average employee salary is over 30K, retrieve the department name and the number of male employees working for it.



```
SELECT DNAME, COUNT (*)
FROM DEPARTMENT, EMPLOYEE
WHERE DNUMBER=DNO AND SEX='M'
GROUP BY DNAME
```

Chapter 4: “More SQL”

b) For each department whose average employee salary is over 30K, retrieve the department name and the number of male employees working for it.

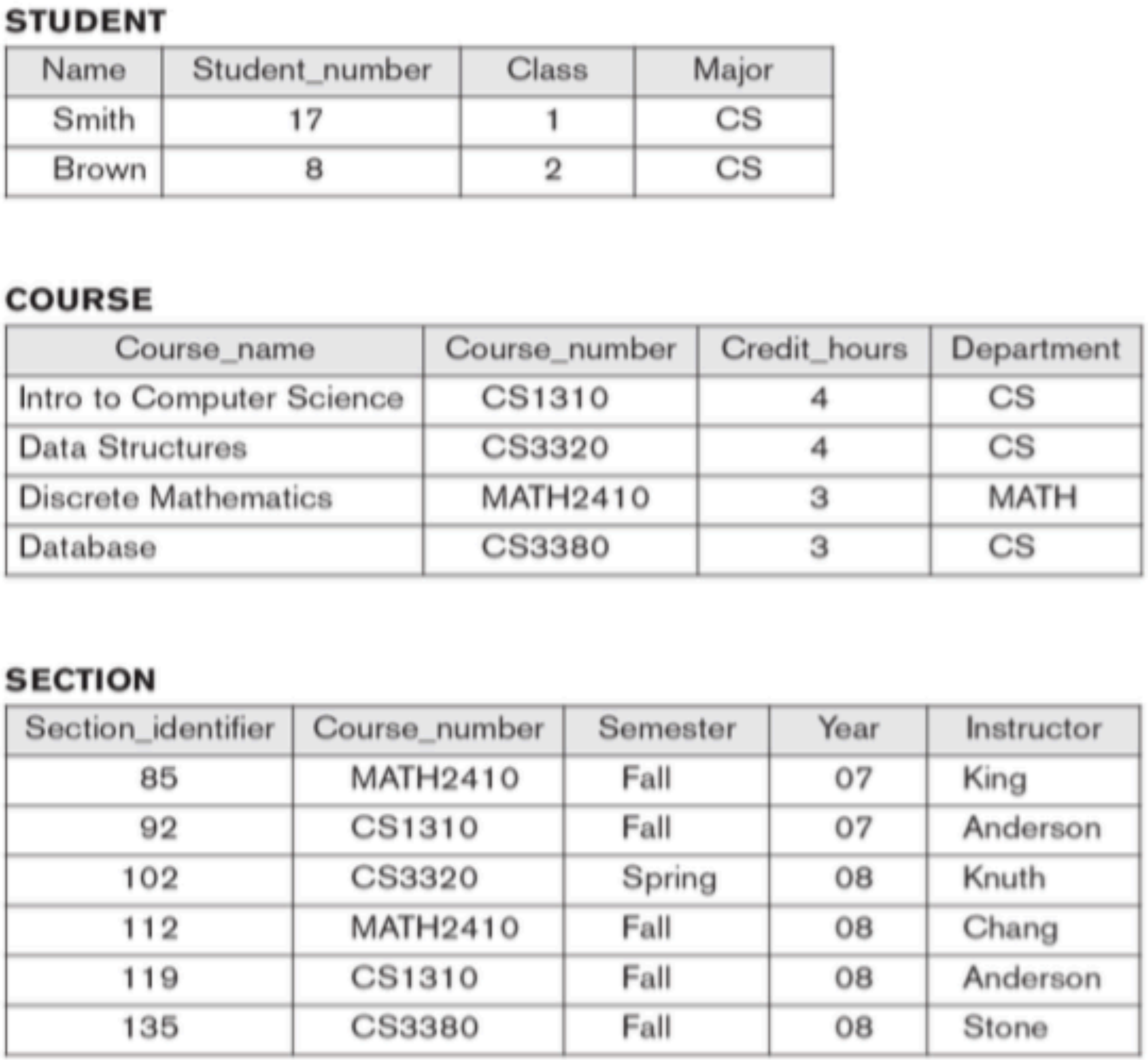


```
SELECT DNAME, COUNT (*)
FROM DEPARTMENT, EMPLOYEE
WHERE DNUMBER=DNO AND SEX='M' AND DNO IN ( SELECT DNO
                                           FROM EMPLOYEE
                                           GROUP BY DNO
                                           HAVING AVG (SALARY) > 30000 )
GROUP BY DNAME
```

Chapter 5: “More SQL”

Exercise 5.6

Specify the following SQL queries on the UNIVERSITY database schema of Fig. 1.2



STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

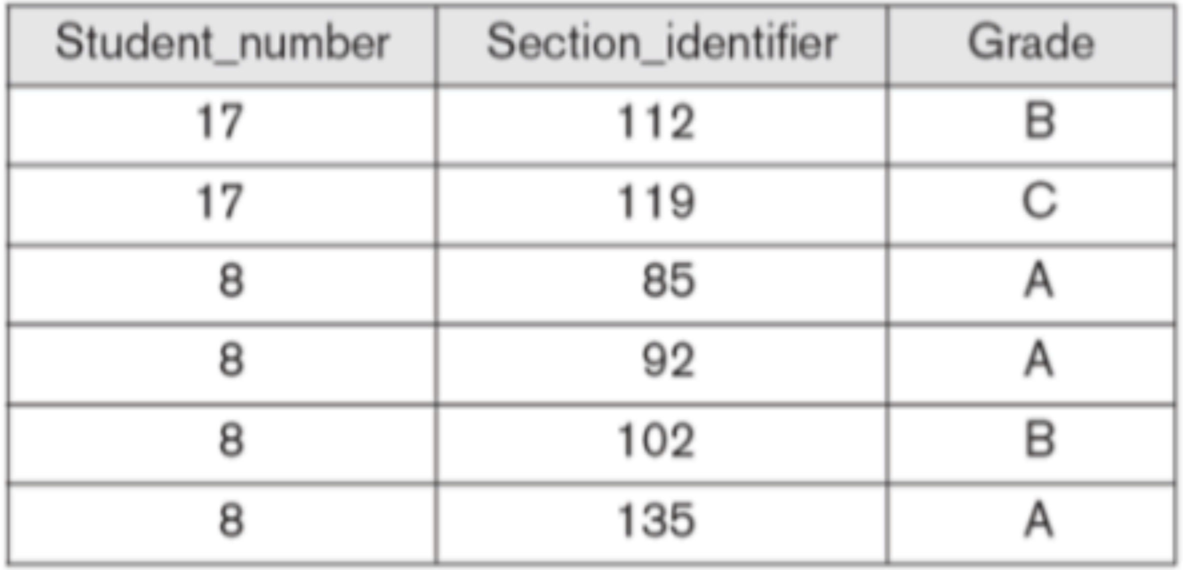
COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

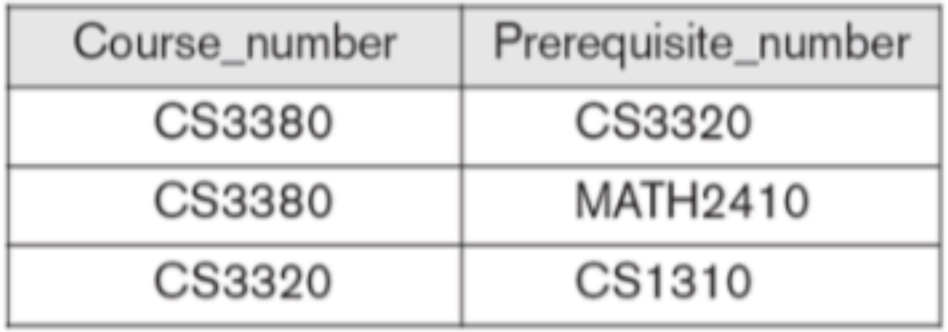
Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE_REPORT



Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE



Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Chapter 5: “More SQL”

Exercise 5.6

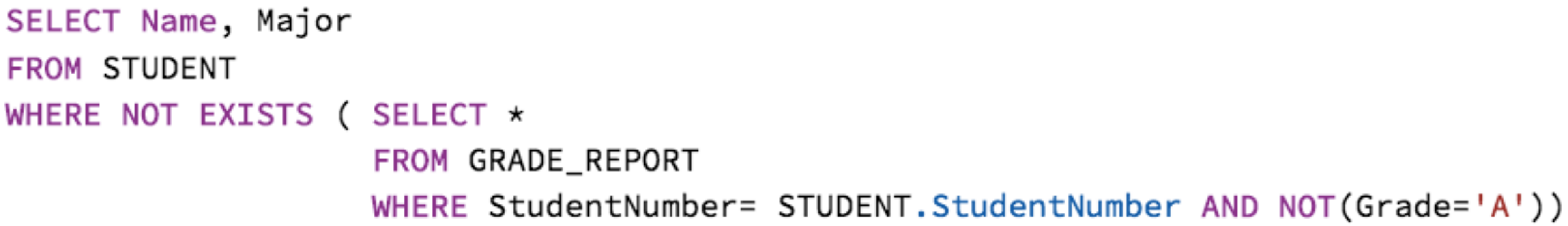
a) Retrieve the names and major departments of all straight-A students (i.e. those who got ‘A’ in all their courses)

Idea: Find all the names of the students such that they received at least one grade that is not A and filter them out of the query.

Chapter 5: “More SQL”

Exercise 5.6

a) Retrieve the names and major departments of all straight-A students (i.e. those who got ‘A’ in all their courses)



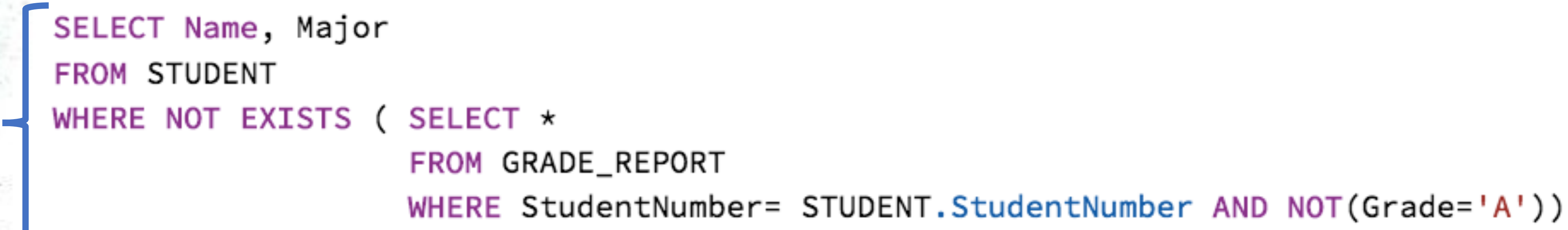
```
SELECT Name, Major
FROM STUDENT
WHERE NOT EXISTS ( SELECT *
                   FROM GRADE_REPORT
                   WHERE StudentNumber= STUDENT.StudentNumber AND NOT(Grade='A'))
```

Retrieve all the columns in GRADE_REPORT such that the student did not receive a grade A

Chapter 5: “More SQL”

Exercise 5.6

a) Retrieve the names and major departments of all straight-A students (i.e. those who got ‘A’ in all their courses)



```
SELECT Name, Major
FROM STUDENT
WHERE NOT EXISTS ( SELECT *
                   FROM GRADE_REPORT
                   WHERE StudentNumber= STUDENT.StudentNumber AND NOT(Grade='A'))
```

Retrieve the names and majors of all the students such that they got an ‘A’ in all their courses

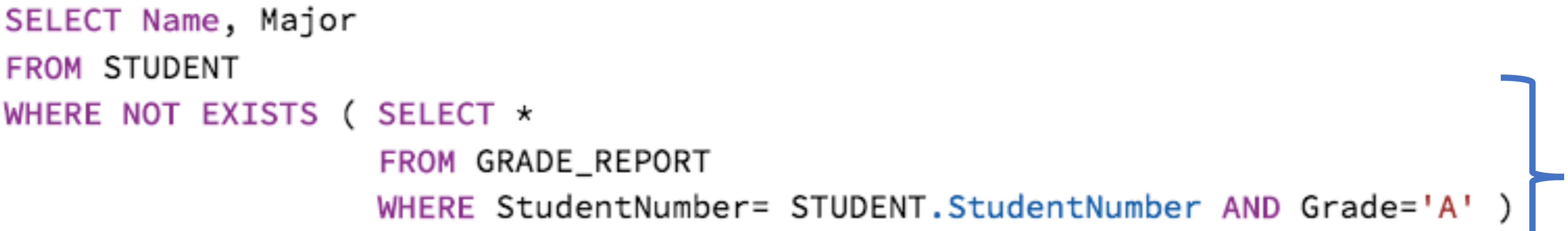
Retrieve all the columns in GRADE_REPORT such that the student did not receive a grade A

Chapter 5: “More SQL”

b) Retrieve the names and major departments of all students who do not have any grade of A in any of their courses.

Chapter 5: “More SQL”

b) Retrieve the names and major departments of all students who do not have any grade of A in any of their courses.

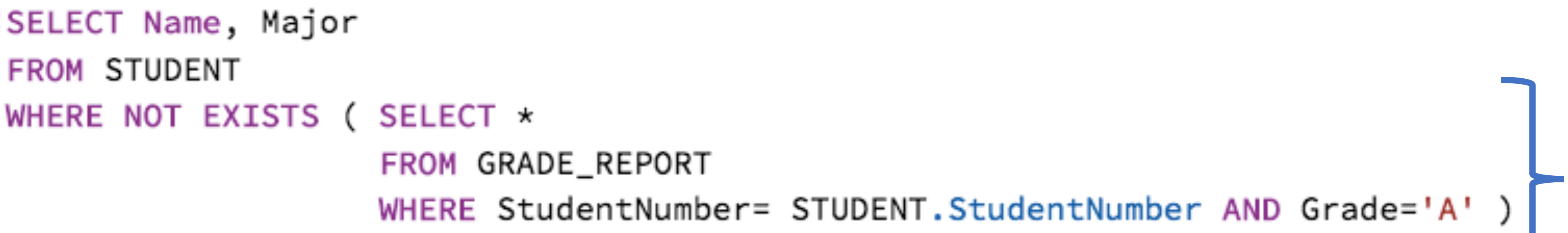


```
SELECT Name, Major
FROM STUDENT
WHERE NOT EXISTS ( SELECT *
                   FROM GRADE_REPORT
                   WHERE StudentNumber= STUDENT.StudentNumber AND Grade='A' )
```

Retrieves all columns from GRADE_REPORT such that the student received a grade of A

Chapter 5: “More SQL”

b) Retrieve the names and major departments of all students who do not have any grade of A in any of their courses.



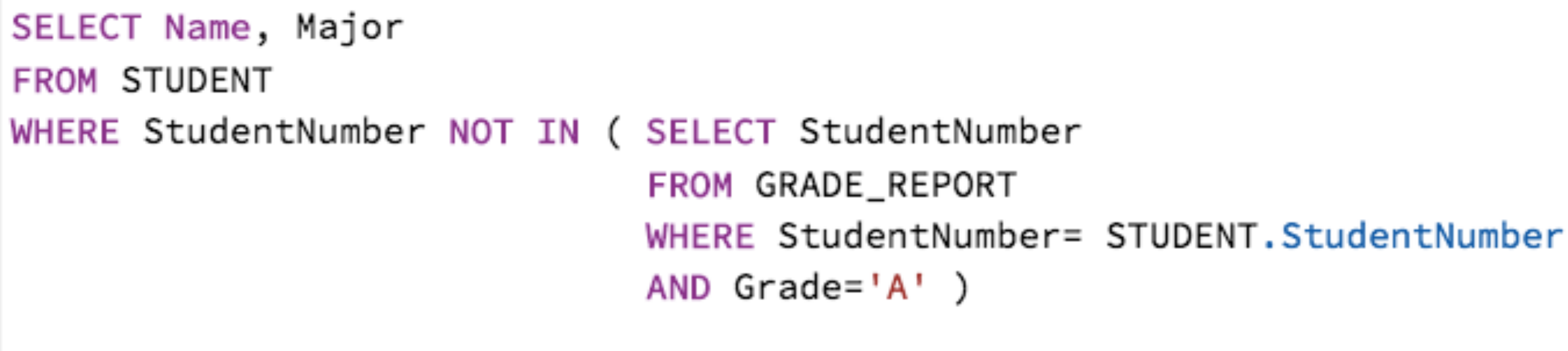
```
SELECT Name, Major
FROM STUDENT
WHERE NOT EXISTS ( SELECT *
                   FROM GRADE_REPORT
                   WHERE StudentNumber= STUDENT.StudentNumber AND Grade='A' )
```

Retrieves the names and majors of all students such that do not have any grade of A in any of their courses

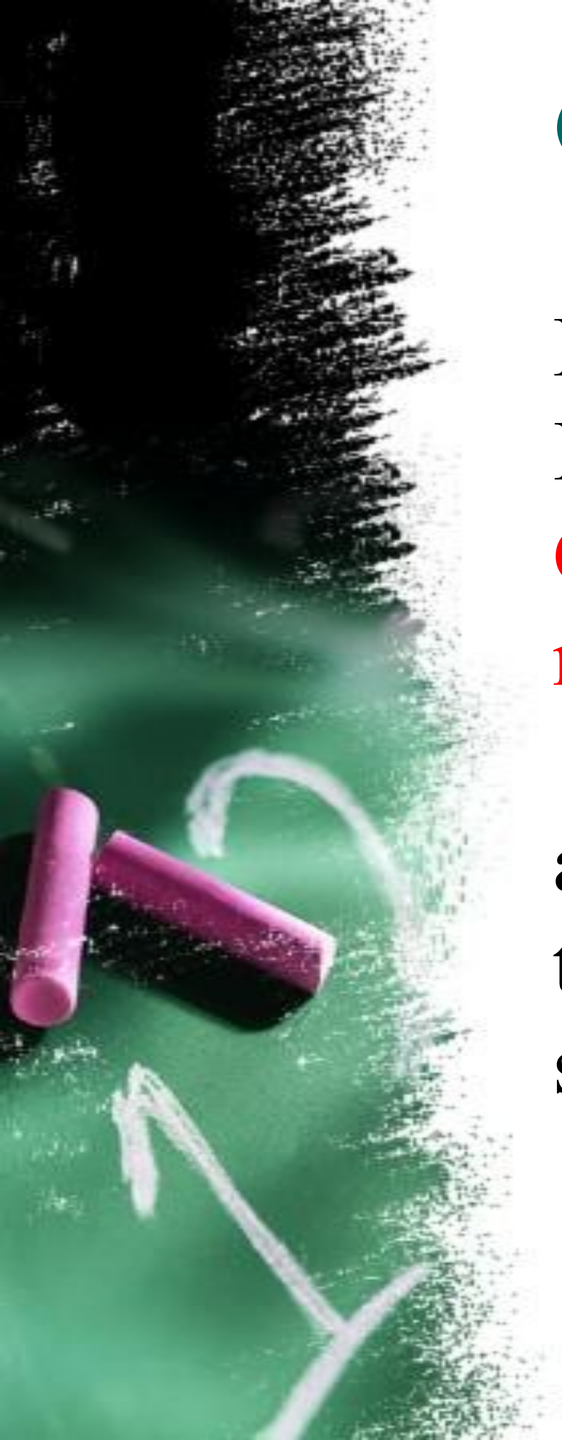
Retrieves all columns from GRADE_REPORT such that the student received a grade of A

Chapter 5: “More SQL”

Another way



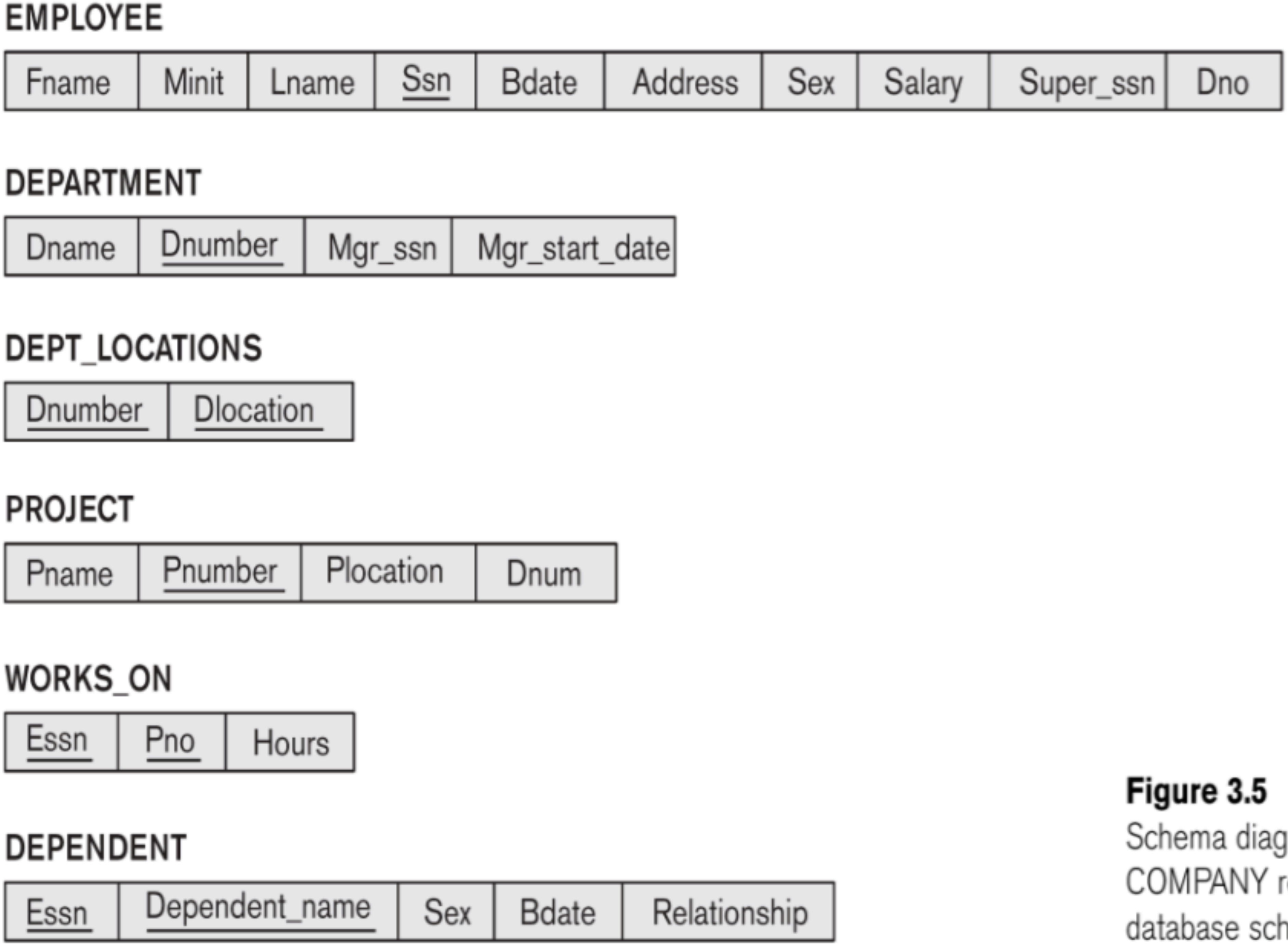
```
SELECT Name, Major
FROM STUDENT
WHERE StudentNumber NOT IN ( SELECT StudentNumber
                             FROM GRADE_REPORT
                             WHERE StudentNumber= STUDENT.StudentNumber
                             AND Grade='A' )
```



Chapter 5: “More SQL”

Exercise 5.7
In SQL, specify the following queries on the COMPANY database in Fig. 3.5 using the concept of nested queries

a) Retrieve the names of all employees who work in the department that has the employee with the highest salary among all employees.



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

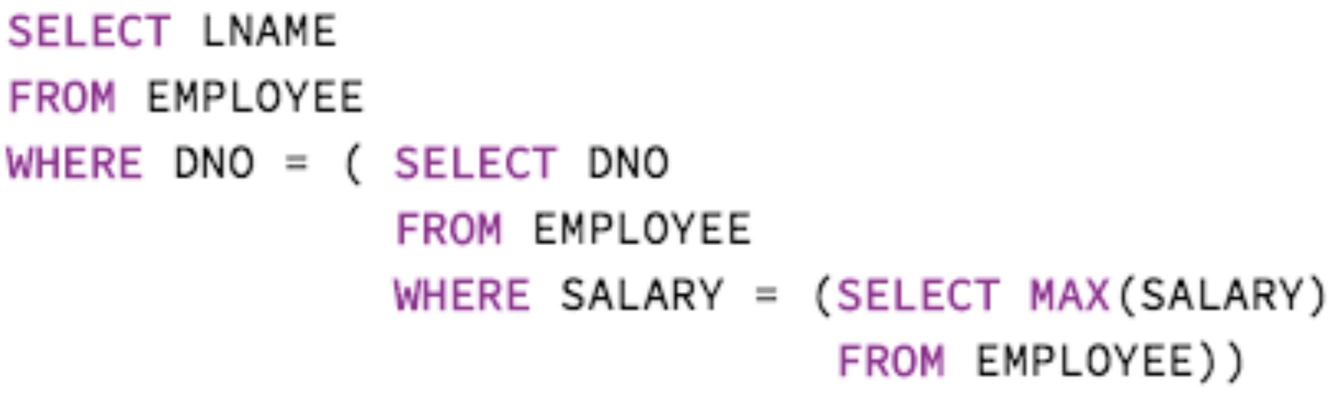
DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Figure 3.5
Schema diagram for the COMPANY relational database schema.

Chapter 5: “More SQL”

a) Retrieve the names of all employees who work in the department that has the employee with the highest salary among all employees.

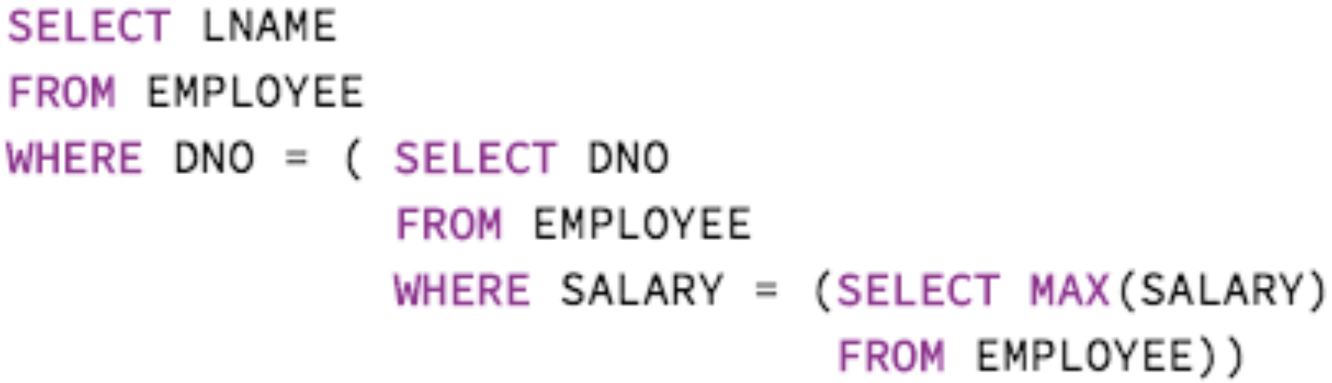


```
SELECT LNAME
FROM EMPLOYEE
WHERE DNO = ( SELECT DNO
              FROM EMPLOYEE
              WHERE SALARY = (SELECT MAX(SALARY)
                              FROM EMPLOYEE))
```

Calculates the
highest salary among
all employees

Chapter 5: “More SQL”

a) Retrieve the names of all employees who work in the department that has the employee with the highest salary among all employees.



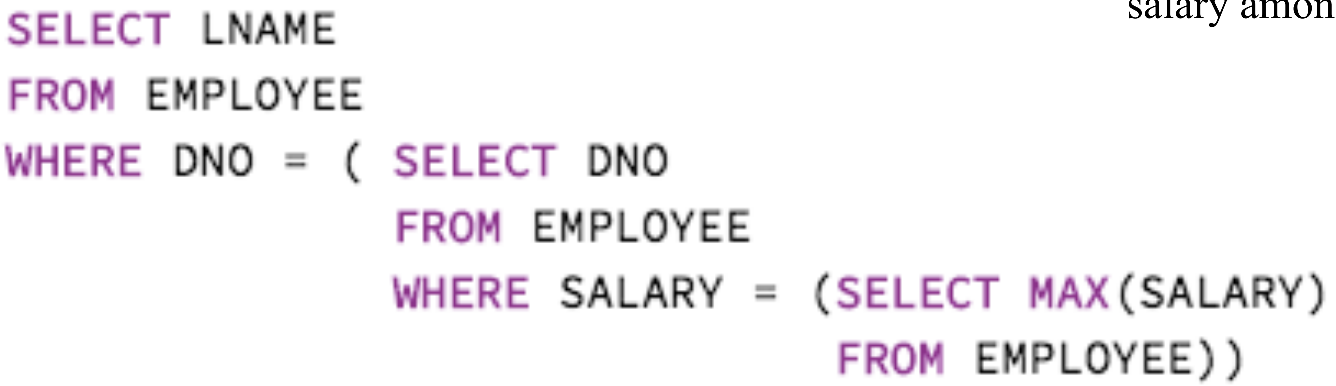
```
SELECT LNAME
FROM EMPLOYEE
WHERE DNO = ( SELECT DNO
              FROM EMPLOYEE
              WHERE SALARY = (SELECT MAX(SALARY)
                              FROM EMPLOYEE))
```

Calculates the highest salary among all employees

Outputs the department number that has the employee with the highest salary

Chapter 5: “More SQL”

a) Retrieve the names of all employees who work in the department that has the employee with the highest salary among all employees.



```
SELECT LNAME
FROM EMPLOYEE
WHERE DNO = ( SELECT DNO
              FROM EMPLOYEE
              WHERE SALARY = (SELECT MAX(SALARY)
                                FROM EMPLOYEE))
```

Retrieve the last names of all employees who work in the department that has the employee with the highest salary among all employees.

Calculates the highest salary among all employees

Outputs the department number that has the employee with the highest salary

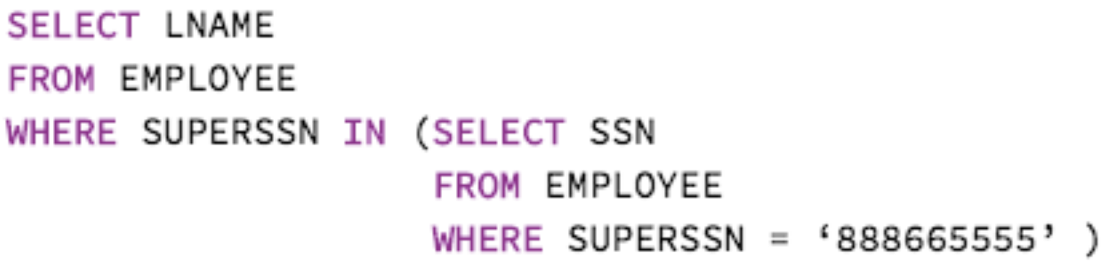
Chapter 5: “More SQL”

Exercise 5.7
b) Retrieve the names of all employees whose supervisor’s supervisor has ‘888665555’ for SSN.

Chapter 5: “More SQL”

Exercise 5.7

b) Retrieve the names of all employees whose supervisor’s supervisor has ‘888665555’ for SSN.



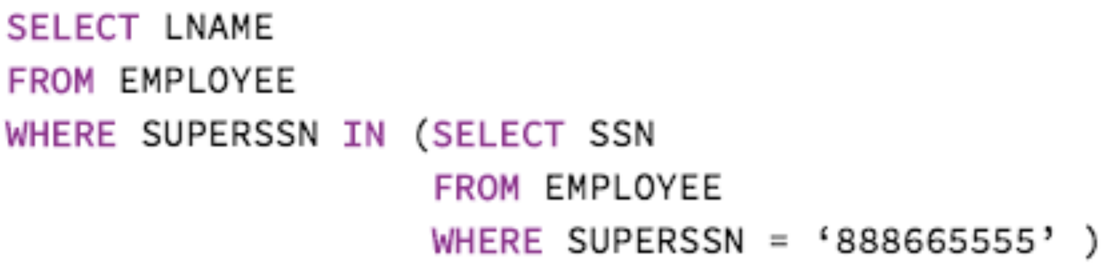
```
SELECT LNAME
FROM EMPLOYEE
WHERE SUPERSSN IN (SELECT SSN
                   FROM EMPLOYEE
                   WHERE SUPERSSN = ‘888665555’ )
```

Retrieve the SSN of all employees who have a supervisor that has ‘888665555’ for SSN

Chapter 5: “More SQL”

Exercise 5.7

b) Retrieve the names of all employees whose supervisor’s supervisor has ‘888665555’ for SSN.



```
SELECT LNAME
FROM EMPLOYEE
WHERE SUPERSSN IN (SELECT SSN
                   FROM EMPLOYEE
                   WHERE SUPERSSN = ‘888665555’ )
```

Retrieve the names of all employees whose supervisor’s supervisor has ‘888665555’ for SSN

Retrieve the SSN of all employees who have a supervisor that has ‘888665555’ for SSN

Chapter 5: “More SQL”

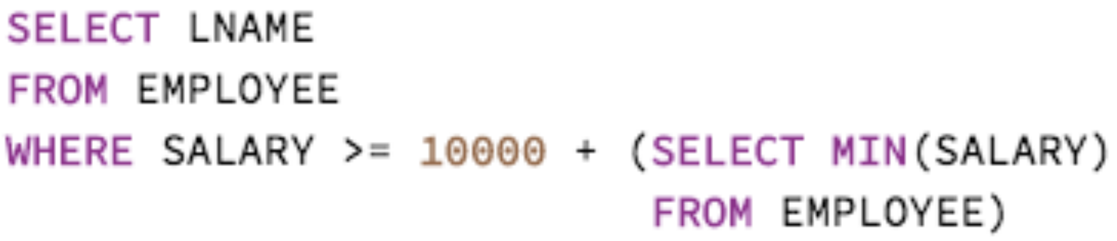
Exercise 5.7

c) Retrieve the names of employees who make at least 10K more than the employee who is paid the least in the company.

Chapter 5: “More SQL”

Exercise 5.7

c) Retrieve the names of employees who make at least 10K more than the employee who is paid the least in the company.



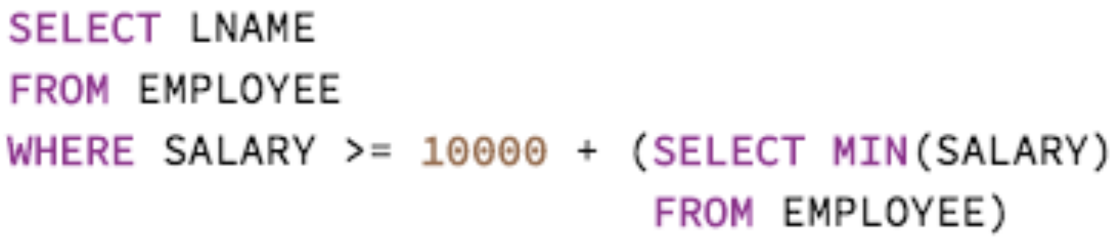
```
SELECT LNAME
FROM EMPLOYEE
WHERE SALARY >= 10000 + (SELECT MIN(SALARY)
                         FROM EMPLOYEE)
```

Retrieves the minimum salary across all employees

Chapter 5: “More SQL”

Exercise 5.7

c) Retrieve the names of employees who make at least 10K more than the employee who is paid the least in the company.



```
SELECT LNAME
FROM EMPLOYEE
WHERE SALARY >= 10000 + (SELECT MIN(SALARY)
                         FROM EMPLOYEE)
```

Retrieves the minimum salary across all employees

Retrieve the last names of employees who make at least 10K more than the employee who is paid the least in the company