COMP-3150 CHAPTER 14: LECTURE NOTES (Ch 14) Covered on Oct. 6, 2020 or Sept. 30, 2021; Oct. 20, 2020 or Oct. 5, 2021; Oct. 22, 2020 or Oct. 7, 2021, Oct. 27, 2020 or Oct. 19, 2021

Chapter 14: Basics of Functional Dependencies and Normalization for Relational Databases

*** Note that Midterm 1 is written in the class of Thursday, Oct. 7, 2021 ~~(Oct. 8, 2020)~~ in online in class. Come to class as usual and log on Blackboard to do your midterm 1 online during the class. The following are the instructions to follow for Midterm 1 so you do not waste too much time going over them during the test:

**

INSTRUCTIONS (Please Read Carefully)

Examination Period is 1 hours 20 minutes (For online version time extended to 2 hours)

Answer all questions. Write your answers in the spaces provided in the question paper. This is closed book and closed notes test. You can type in your answers into the word file and submit, or print, write with hand, scan clearly into only a .pdf or .jpeg file and submit.

Total Marks =50. Total number of sections = 2

Please read questions carefully! Misinterpreting a question intentionally or unintentionally results in getting a "ZERO" for that question. Good Luck!!!

CONFIDENTIALITY AGREEMENT & STATEMENT OF HONESTY

I confirm that I will keep the content of this assignment/examination confidential.

I confirm that I have not received any unauthorized assistance in preparing for or doing examination. I confirm knowing that a mark of 0 may be assigned for copied work.

For Online Test/Examination in Comp 3150 Fall 2020: (additional rules to be observed):

1. I confirm that I agree to write this final examination as a closed book examination.

2. I confirm that I am the student with the name and student id signed below.

3. I confirm that I agree to not send email, chat, text or talk in any way to people other than the instructor or proctoring GA of this course during this examination.

4. I confirm that I agree to not engage in copying or cheating during this online examination.

_____ _____

Student Signature Student Name (please print)

_____ _____

Student I.D. Number Date

_____

NOTE THAT I WILL BE GOING OVER IN THE LAST 10 MINUTES OF A CLASS YOUR ANSWERS TO THE MIDTERM 1 PRACTICE TEST I POSTED FOR YOU. NO FORMAL SOLUTIONS TO ANY PRACTICE TEST IS POSTED ON THE WEB. HOWEVER, IT WILL BE REVIEWED WITH YOU IN THIS CLASS. PARTS OF THE SOLUTION MAY ALSO BE FOUND AT THE END OF THIS LECTURE NOTES.

Fall 2021 Comp 3150 online recordings can be downloaded from the black board virtual classroom for the class of the day. The links below are the recordings of similar classes in Fall 2020 that I uploaded to one drive in case you prefer to review these.

Links to any posted recorded in-class lectures for Chapter 14 Oct. 6, 2020 (similar to Sept. 30, 2021), Oct. 20, 2020 (similar to Oct. 5, 2021) and Oct. 22, 2020 (similar to Oct. 7, 2021) will be found below:

1. Links on one drive to recorded lecture of Tuesday, Oct. 6, 2020 (similar to Sept. 30, 2021) (saved in two recorded sessions to break down the files) is provided below in sequences 1, 2:

Frist Recording 1 of 2:

https://uwin365-my.sharepoint.com/:v:/g/personal/cezeife_uwindsor_ca/Ebmj9s1B5gpJv88g2pD3qRUBApRocu9N627AWUsTYrEU1w

Second Recording 2 of 2:

https://uwin365-my.sharepoint.com/:v:/g/personal/cezeife_uwindsor_ca/EbdGh4o0U3FNuTNe4hW1lF8BxM3hf-mGCmfCtEwuPd_NMg

2. Links on one drive to recorded lecture of Thursday, Oct. 8, 2020 (similar to Oct. 5, 2021) (Midterm 1 test in BB) (saved in two

recorded sessions to break down the files) is provided below in sequences 1, 2:

Frist Recording 1 of 2:

https://uwin365-my.sharepoint.com/:v:/g/personal/cezeife_uwindsor_ca/EU-a8OjJ0GdFkdxjvJIKLpsBd3kSIfAFBC5g2sgd7PZSZQ

Second Recording 2 of 2:

https://uwin365-my.sharepoint.com/:v:/g/personal/cezeife_uwindsor_ca/EYBAaCzYaopFv5atZNzvQS0ByrXH_yPQk5KbXCarOo5ryw

3. Links on one drive to recorded lecture of Thursday, Oct. 20, 2020 (similar to Oct. 7, 2021) (saved in two recorded sessions to break down the files) is provided below in sequences 1, 2:

Frist Recording 1 of 2:

https://uwin365-my.sharepoint.com/:v:/g/personal/cezeife_uwindsor_ca/ETEpagI9jxVFoJyiavnbvD4B5QTPxhO1vhkneRSGlqqmfw

Second Recording 2 of 2:

https://uwin365-my.sharepoint.com/:v:/g/personal/cezeife_uwindsor_ca/ER-gFIORQa5PnrxuvthfTcoBgD_ucXHMR5L9TV1fisAG8Q

4. Links on one drive to recorded lecture of Thursday, Oct. 22, 2020 (similar to Oct. 19, 2021) (saved in two recorded sessions to break down the files) is provided below in sequences 1, 2:

Frist Recording 1 of 2:

https://uwin365-my.sharepoint.com/:v:/g/personal/cezeife_uwindsor_ca/EVo-NaBks6aJJtXWEhn26wSQBUrqVazdn5m8deoEN-BEd2A

Second Recording 2 of 2:

https://uwin365-my.sharepoint.com/:v:/g/personal/cezeife_uwindsor_ca/EcWiez_NBbRGteZSJn3JA2QBSwnfULat9J4EoDvamu35mQ

5. Links on one drive to recorded lecture of Tuesday, Oct. 27, 2020 (similar to Oct. 21, 2021) (saved in two recorded sessions to break down the files) is provided below in sequences 1, 2:

Frist Recording 1 of 2:

https://uwin365-my.sharepoint.com/:v:/g/personal/cezeife_uwindsor_ca/EWLfQT-pHONBmVxRKKV9KxoBqSy4lMbfHC4kiO9XNG6V5A

Second Recording 2 of 2:

https://uwin365-my.sharepoint.com/:v:/g/personal/cezeife_uwindsor_ca/EZ62-Z3SM81NmP5XaOFW_uEB725Ok9UKmL5FEe8sSBEB8A

***

The following questions on Basics of Functional Dependencies and Normalization for Relational Databases discussed in Chapter 14 of Comp 3150 text book, Chapter 14, Comp 3150 posted course slide notes, are in-class questions for students to ponder and answer as I teach.

- The answers to the questions are found also by reviewing the Comp 3150, posted power point slide notes for Chapter 14 and being in class.

- Students are advised to review Chapters 14 of course book and Comp 3150 posted slide notes before and after each class.

- I will also go over the Slide notes in class with examples and integrate them into the class lectures, which are also posted in the More course material link on black board with any links to recorded live lectures.

1. HOW DO WE KNOW THAT OUR DATABASE DESIGN IS GOOD AND WOULD WORK WELL?

- The course book specifies 4 informal design guidelines used to measure the quality of relation schema design.

i. Attributes in the schema have clear semantics. Guideline 1: Design a relation with clear meaning and which does not combine attributes from multiple entity types such as Employee and Department. (See an example

of this combination below and in Ch. 14 slides).
** The problem is: Storing relations that combine attributes from more than one entity type would lead to anomalies (which are insertion, deletion and modification (update) anomalies). Anomalies indicate presence of redundant storage of information in the DB design.

ii. Redundant information in tuples are reduced.
GUIDELINE 2: Design a schema that does not suffer from the insertion, deletion and update anomalies. (See examples of what are these anomalies below and the Ch. 14 slides).
** Problem is: presence of these anomalies in a relation creates redundancies in the database that waste storage and make data sometimes incorrect and hard to update (e.g. Figs. 14.3 and 14.4 on Slides 12 and 13 obtained from a type of natural join of relations in Fig. 14.1 and 14.2).

iii. NULL values in tuples are reduced.
GUIDELINE 3: Relations should be designed such that their tuples will have as few NULL values as possible. Attributes that are NULL frequently could be placed in separate relations (with the primary key).

- For example, in a University DB, rather than have an entity type called Person (SSN, Name, officePhone), where only about 6% of the 15,000 tuples are staff and faculty who have office phone numbers and the rest are students who do not and have NULL values, it is better to design separate entity types for the same information such as:
STUDENT(SID, Name),
STAFF(STID, SNAME, OFFICEPHONE).

iv. Possibility of generating spurious tuples are disallowed.
GUIDELINE 4: The relations should be designed to satisfy the lossless join condition.
- No spurious (fake) tuples (not in the original relation) should be generated by doing a natural-join of any decomposed relations.
- Design relation schemas that can be joined with equality conditions on attributes that are (primary key, foreign key) pairs.

- There are two important properties of decompositions of relations to keep the design good:
  a) Non-additive or losslessness of the corresponding join (that is, not introducing new attributes or tuples and not

losing existing attributes or tuples when decomposed relations are joined back).

b) Preservation of the functional dependencies (FDs) (that is, the FDs holding in the bad relations in the whole DB before decomposition, should be holding in the whole DB after decomposition but each relation should be free of anomalous or violating FDs).

- An example of decomposition of a relation R that yields spurious tuples is given below.

- Let R called Properties(<u>Ssn</u>, books, pens) be the original relation and let this relation be decomposed into two relations Properties1(Ssn, books) and Properties2(book, pens)

(a) Original relation Two decomposed relations (b) and (c)

Properties		
Ssn	Books	Pens
1	12	8
2	25	5

Properties1	
Ssn	Books
1	12
2	25

Properties2	
Books	Pens
12	8
25	5

Result of joining back decomposed relations Properties1 and Properties2

Properties1 join Properties2

Ssn	Books	Pens	
1	12	8	
1	25	5	
2	12	8	
2	25	5	** The spurious tuples not in the original relation Properties are the 2 highlighted middle tuples.

- A correct decomposition of Properties is shown below:

(a) Original relation Two decomposed relations (b) and (c)

Properties			Properties3		Properties4	
Ssn	Books	Pens	Ssn	Books	Ssn	Pens
1	12	8	1	12	1	8
2	25	5	2	25	2	5

Result of joining back decomposed relations Properties3 and Properties4

Properties3 join Properties4

Ssn	Books	Pens	
1	12	8	
2	25	5	** No spurious tuples not in the original relation Properties are generated here.

Note that:

Property (a) is extremely important and cannot be sacrificed. Property (b) is less stringent and may be

sacrificed.

2. WHAT ARE FUNCTIONAL DEPENDENCIES?

3. HOW DO WE DECOMPOSE RELATIONS INTO NORMALISED RELATIONS THAT ARE IN THE 1ST NORMAL FORM, 2ND NORMAL FORM, 3RD NORMAL FORM, BOYCE CODD NORMAL FORM (BCNF)?

(ALSO, 4TH AND 5TH NORMAL FORM EXIST But not addressed in this course).

- Note that a well-designed normalized relation needs to be in at least the 3rd normal form.

A fast example on normal forms:

Given the relation

Takes (Sid, Cid, Score, Grade)

Is this relation in 3NF?

- A relation is in 3NF if only the whole of the primary key uniquely determine all other attributes and no partial or transitive dependencies exist.

The functional dependencies that exist in this Takes are:

FD1: (Sid, Cid) → {Score, Grade} // because it is the PK

FD2: Score → Grade // this is transitive dependency

- Takes is NOT in 3NF because of FD2 that violates the 3NF rule.

- Thus, Takes contains anomalies.

- Can we decompose Takes to keep the database in 3NF?

- To put the DB in 3NF, create new tables with each violating FD. And remove all the right hand attributes in the violating FDs from the original DB table, which is also kept as a table in the DB.

- The normalized database form of Takes above are:

  Takes1 (Sid, Cid, Score)
  Score1(Score, Grade)

The functional dependencies that exist in this new normalized DB are:

FD1: (Sid, Cid) → {Score} // because it is the PK
// Takes1

FD2: Score → Grade // for PK in Score1

- Thus, although the FDs in the 3NF DB are the same as those in the non-normalized table Takes, those FDs are in different normalized tables in Takes1 and Score1.
  ***

Examples Describing Guideline 1: Combination of Entity types in one relation.

Slide 6 of Course Slide Notes of Chapter 14 shows Fig. 14.1, a simplified COMPANY relational DB schema. A state of this DB is Fig. 14.2.

Figure 14.1 A simplified COMPANY relational database schema.

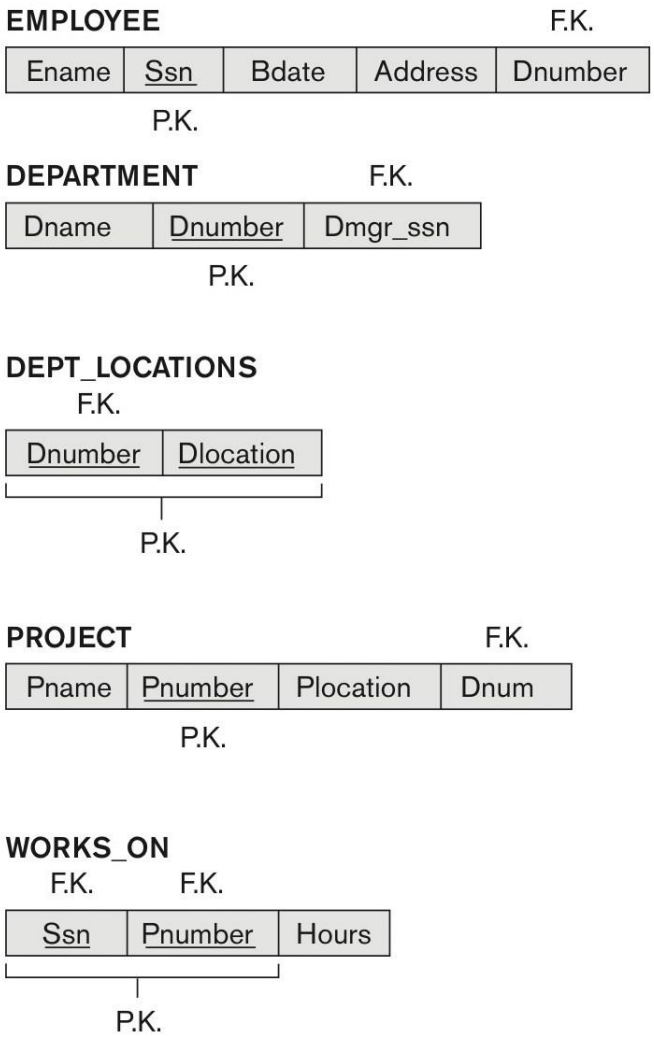


Figure 14.2 Sample database state for the relational database schema in Figure 14.1.

EMPLOYEE

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

DEPARTMENT

Dname	Dnumber	Dmgr_ssn
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

DEPT_LOCATIONS

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

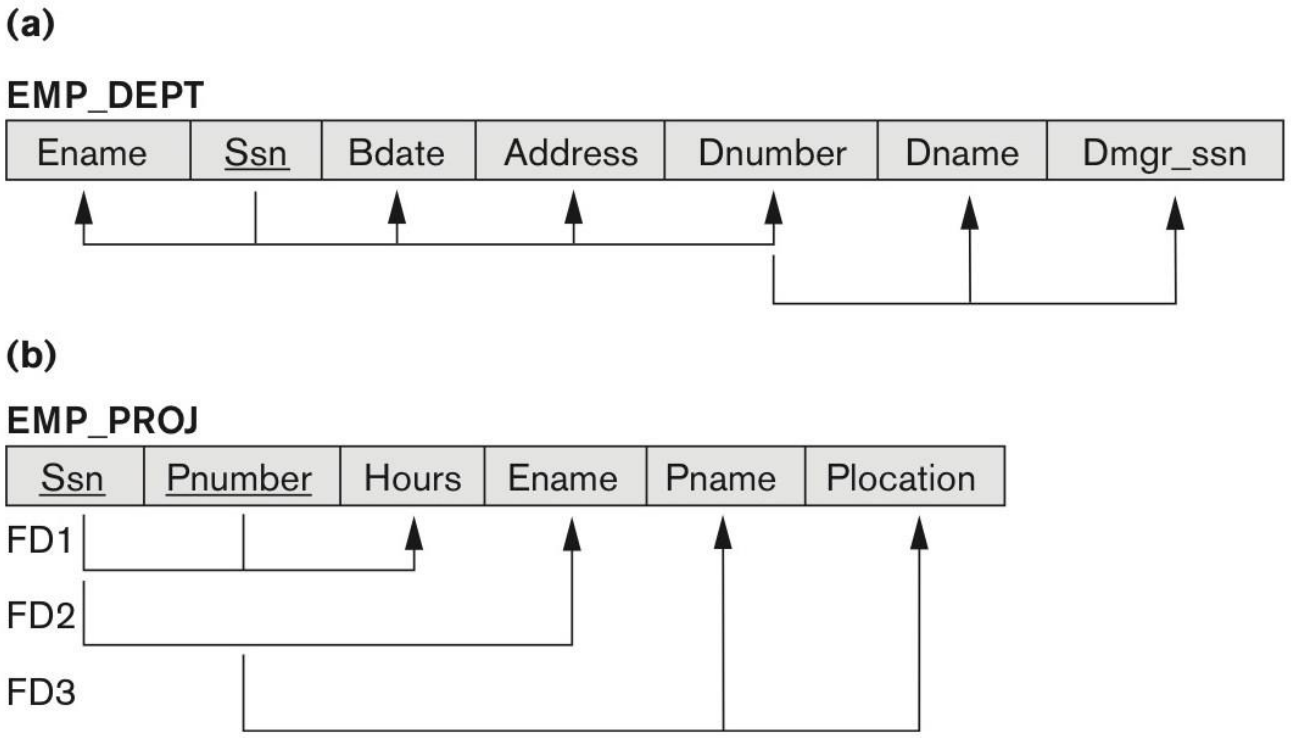
WORKS_ON

Ssn	Pnumber	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	Null

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

Figure 14.3 Two relation schemas suffering from update anomalies. (a) EMP_DEPT and (b) EMP_PROJ.



***

Figure 14.4 Sample states for EMP_DEPT and EMP_PROJ resulting from applying NATURAL JOIN to the relations in Figure 14.2. These may be stored as base relations for performance reasons.

Redundancy

EMP_DEPT

Ename	Ssn	Bdate	Address	Dnumber	Dname	Dmgr_ssn
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston, TX	5	Research	333445555
Wong, Franklin T.	333445555	1955-12-08	638 Voss, Houston, TX	5	Research	333445555
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle, Spring, TX	4	Administration	987654321
Wallace, Jennifer S.	987654321	1941-06-20	291 Berry, Bellaire, TX	4	Administration	987654321
Narayan, Ramesh K.	666884444	1962-09-15	975 FireOak, Humble, TX	5	Research	333445555
English, Joyce A.	453453453	1972-07-31	5631 Rice, Houston, TX	5	Research	333445555
Jabbar, Ahmad V.	987987987	1969-03-29	980 Dallas, Houston, TX	4	Administration	987654321
Borg, James E.	888665555	1937-11-10	450 Stone, Houston, TX	1	Headquarters	888665555

Redundancy Redundancy

EMP_PROJ

Ssn	Pnumber	Hours	Ename	Pname	Plocation
123456789	1	32.5	Smith, John B.	ProductX	Bellaire
123456789	2	7.5	Smith, John B.	ProductY	Sugarland
666884444	3	40.0	Narayan, Ramesh K.	ProductZ	Houston
453453453	1	20.0	English, Joyce A.	ProductX	Bellaire
453453453	2	20.0	English, Joyce A.	ProductY	Sugarland
333445555	2	10.0	Wong, Franklin T.	ProductY	Sugarland
333445555	3	10.0	Wong, Franklin T.	ProductZ	Houston
333445555	10	10.0	Wong, Franklin T.	Computerization	Stafford
333445555	20	10.0	Wong, Franklin T.	Reorganization	Houston
999887777	30	30.0	Zelaya, Alicia J.	Newbenefits	Stafford
999887777	10	10.0	Zelaya, Alicia J.	Computerization	Stafford
987987987	10	35.0	Jabbar, Ahmad V.	Computerization	Stafford
987987987	30	5.0	Jabbar, Ahmad V.	Newbenefits	Stafford
987654321	30	20.0	Wallace, Jennifer S.	Newbenefits	Stafford
987654321	20	15.0	Wallace, Jennifer S.	Reorganization	Houston
888665555	20	Null	Borg, James E.	Reorganization	Houston

```
SQL> select ssn, pno, fname, lname, pname, hours
 2  from employee, works_on, project
 3  where ssn = essn and pno = pnumber;
```

SSN	PNO	FNAME	LNAME	PNAME	HOURS
999887777	10	Alicia	Zelaya	Computerize	10
987987987	10	Ahmad	Jabbar	Computerize	35
333445555	10	Franklin	Wong	Computerize	10
999887777	30	Alicia	Zelaya	Nbenefits	30
987987987	30	Ahmad	Jabbar	Nbenefits	5
987654321	30	Jennifer	Wallace	Nbenefits	20
453453453	1	Joyce	English	ProductX	20
123456789	1	John	Smith	ProductX	32.5
453453453	2	Joyce	English	ProductY	20
333445555	2	Franklin	Wong	ProductY	10
123456789	2	John	Smith	ProductY	7.5

SSN	PNO	FNAME	LNAME	PNAME	HOURS
666884444	3	Ramesh	Narayan	ProductZ	40
333445555	3	Franklin	Wong	ProductZ	10
987654321	20	Jennifer	Wallace	Reorganize	15
888665555	20	James	Borg	Reorganize	10
333445555	20	Franklin	Wong	Reorganize	10

16 rows selected.

What are the Meanings of the Anomalies in Guideline 2? Explained in Slides 8 to 15.

Consider the relation:

EMP_PROJ(Emp#, Proj#, Ename, Pname, No_hours)

i. Update Anomaly: Changing the name of project number P10 from “Computerize” to “Customer-Accounting” may

cause this update to be made for all 3 (may have more in a bigger state) employees working on project P10.

ii. Insert Anomaly: Cannot insert a project unless an employee is assigned to it. Conversely, Cannot insert an employee unless he/she is assigned to a project. This is because both Emp#, Proj# make the primary key and none of these attributes of the primary key is allowed to be NULL.

iii. Delete Anomaly: When a project is deleted, it may result in deleting all the employees who work on that project. Alternatively, if an employee is the sole employee on a project, deleting that employee would result in deleting the corresponding project.

2. What are Functional Dependencies?

- Functional dependencies (FDs) are used to specify formal measures of the "goodness" of relational designs.

- FDs and keys are used to define normal forms for relations

- FDs are constraints that are derived from the meaning and inter-relationships of the data attributes.

- A set of attributes X functionally determines a set of attributes Y if the value of X determines a unique value for Y

- An example FD in a Student Information system with table TAKES (SID, CID, SCORE, GRADE) is Score →Grade. This means that in this world, you really cannot say what Grade a student has received in a course if you do not have the Score they received, interpreted as: Score determines Grade.

- However, also by their role as the unique identifier, the primary key (consisting of (SID, CID) also determines every tuple and each attribute of the relation. Thus, the following FD also holds in the table TAKES:
(SID, CID) → {Score, Grade}
This means that (SID, CID) determines all other attributes in the table, Score and Grade.

Consider a state or instance of the table TAKES as:
TAKES

SID	CID	SCORE	GRADE
1	COMP2120	80	A-
2	COMP2120	85	A
3	COMP3150	80	A-
4	COMP3150	85	A
5	COMP3150	77	B+

-Only the primary key should be the unique determinant of other non-key attribute or you may have insert, delete or update anomalies.

- We can see that these anomalies exist in TAKES because the FDs existing in this DB are:
FD1: (SID, CID) → {Score, Grade}
FD2: SCORE →Grade

-Note that FD2 is a violating FD in this table that is causing

these anomalies and this should be handled through decomposition to put this table in at least 3NF to have it normalized.

Question:

i. Show how there is insert anomaly in this table TAKES.

-Although it is not the primary key, we cannot insert a student record taking this course with a NULL score in this table. We must wait until they receive a score in the course to indicate the course they have taken. This is not proper as this table should be able to record all students and courses, and grades they received in the course, even if they did not complete this course and have no grade. Thus, we cannot insert a tuple like <6, 'COMP2120', NULL, NULL>.

ii. Show how there is delete anomaly in this table TAKES.

The only way we can tell the association between Score and Grade values are by checking similar values in the table TAKES where for any two tuples (t1 and t2) of TAKES, if t1.Score = t2.Score, then, t1.Grade = t2.Grade. For example, tuples 1 and 3 have their Score = 80 and their grades = 'A-'.

Thus, for delete anomaly, if we happen to delete the last tuple with this score value by deleting both tuples 1 and 3 from the

database, we no longer know the association between the score of 80 and its equivalent letter grade.

iii. Show how there is update anomaly in this table TAKES
If you have one hundred students who received a grade of 80 for example, and the grade scale recently changed from 80 to 83 to be the minimum for a grade of "A-", we must change all 100 records from 80 to 83 or the FD that must exist in the database is violated. This is an example of update anomaly.

iv. Normalize this table TAKES to have it in 3NF and remove these anomalies so you create a normalized database where all of its tables are normalized and the FDs in this TAKES will be preserved in the new normalized DB with no loss of tuples/attributes if joined back.

- The decomposition is better done using the FDs and the rules about 1NF, 2NF and 3NF as discussed on slides 24 to 45 of Ch. 14 slide notes.

- An informal summary of 1NF, 2NF and 3NF rules as on slide 39 is given below:

Normal Forms Defined Informally

- 1st normal form (1NF)
    - All attributes depend on the key and are single valued.
- 2nd normal form (2NF)
    - All attributes depend on the whole key (ie, all attributes are fully functional dependent on the whole key - that is, no partial dependency).
- 3rd normal form (3NF)
    - All attributes depend on nothing but the key (i.e., no attribute should depend on the key through another attribute: no transitive dependency).

TO NORMALIZE TAKES RELATION

1. Identify all existing FDs in the relation including the FD imposed by the primary key, and identify the FDs that are violating any of the 1NF, 2NF or 3NF rules.

The existing FDs in TAKES are:

FD1: (SID, CID) → {Score, Grade} **(FD imposed by PK)

FD2: SCORE →Grade **(violating FD)

2. Remove all right hand attributes of the violating FDs from the right hand side of the main PK FD, while still retaining the violating FD in the DB. This will remove all the violations from the tables and FDs. The result is:

FD1: (SID, CID) → {Score, ~~Grade~~}

FD2: SCORE →Grade.

3. Convert each of the FDs into a normalized DB table with a unique name where the primary key of the table from the FD is the left side attributes of the FD and the rest of the attributes on its right side are the other attributes as in:

TAKES (SID, CID, Score)

SCOREINFO (SCORE, Grade)

The FDs in the above normalized DB with two tables are the two FDs refined above as:

FD1: (SID, CID) → {Score}

FD2: SCORE →Grade.

Ch 14: More on Normal Forms: Slide 36 of Ch 14 Slides

Is this relation in 2NF?

Emp_Proj(Ssn, Pnumber, Hours, Ename, Pname, Plocation)

FDs:
FD1: (Ssn, Pnumber)—> {Hours, Ename, Pname, Plocation}
FD2: Ssn → Ename (Violating)
FD3: Pnumber → Pname, Plocation (Violating)

2NF: A relation R is in 2NF if every non-key attribute A in R is FFD (fully functionally dependent) on the primary key (Pk) (and not on any proper subset pf the PK).

** Note ~~FFD~~ (struck out) here means does not fully functionally determine.

Emp_Proj is not in 2NF because the Pk (Ssn,Pnumber) ~~$\overrightarrow{FFD}$~~ Ename because of FD2. In FD2 Ssn is a subset of the Pk and Ssn→ Ename.
Also, the presence of FD3 shows that
(Ssn,Pnumber) ~~$\overrightarrow{FFD}$~~ Pname, Plocation

Normalize Emp_Proj into 2NF. Normalization is usually accomplished through decomposition of a relation to remove the violating FD's from one single table.

Emp_Proj
FD1: (Ssn, Pnumber) → {Hours, ~~Ename~~, ~~Pname~~, ~~Plocation~~}
Remove the right hand side attributes of violating FD's (FD2 and FD3). Then, re-insert the updated FD1 with the violating FD's in the DB and derive relations from all these FD's to have your normalized DB.

FD's
FD1: (Ssn, Pnumber)—> Hours
FD2: Ssn → Ename
FD3: Pnumber → {Pname, Plocation}

Create the tables from FD's where the schema consists of all attributes in each FD with left hand side attributes as the PK.

Emp_proj1 (Ssn, Pnumber, Hours)
Emp_Proj2 (Ssn, Ename)
Emp_Proj3 (Pnumber, Pname, Plocation)

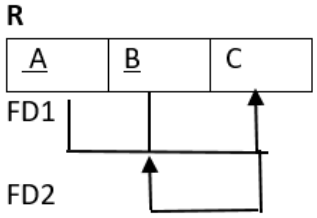
Ch. 14 Continues: BCNF: Slides 41 to 44

1. An example of relation which is in 3NF but not in BCNF.

   BCNF differs from the 3NF in definition because in the general definition of 3NF, as clause (b) mentioned on slide 41 allows FD's that have the right hand side of X → A as a prime (key) attribute (that is determined by a non-key attribute), is not allowed in the BCNF.

   For example:

   R (A, B, C), here R is a relation in 3NF but not in BCNF.
   FD1: (A, B) → C
   FD2: C → B (BCNF violation)



   FD2 violates the BCNF, but R is in 3NF but not in BCNF.

2. How can we put R in BCNF?

   In order to normalize R into BCNF, we decompose R into BCNF tables by first removing the violating FD from the R the same way we did with decomposition of non-normalized tables into 2NF and 3NF.

   Normalized FDs are:
   FD1: (A, ~~B~~) → C (removing right hand side of FD2 from FD1)
   FD2: C → B (BCNF violation) (needs to be preserved as before)

   Resulting BCNF tables arising from the FD1 and FD2 above are:

R1 (A, C)
R2 (C, B)

R1 and R2 are in BCNF. The Functional dependencies are as follows:

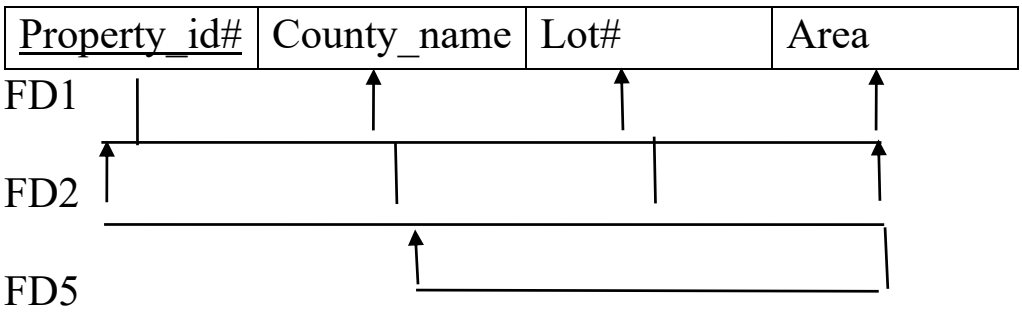
FD1: A → C
FD2: C → B

—

An Example of BCNF used in the book on Figs. 14.11 to 14.13 for normalizing the LOTS1A relation into BCNF.

3. Boyce-Codd normal form. BCNF normalization of LOTS1A with the functional dependency FD2 being lost in the decomposition. Note that (County_Name, Lot#) is a candidate key.

LOTS1A



FD1: Proprty_id# → County_name, Lot#, Area
FD2: County_name, lot# → Property_id#, Area
FD5: Area → County_name (violates BCNF)
(County_name, Lot#) is a candidate key.

LOTS1A is in 3NF but not in BCNF. To decompose it into BCNF, remove the FD5 from the main table and re-insert it as a separate table.

After BCNF Normalization

LOTS1AX

Propoerty_id#	Lot#	Area

LOTS1AY

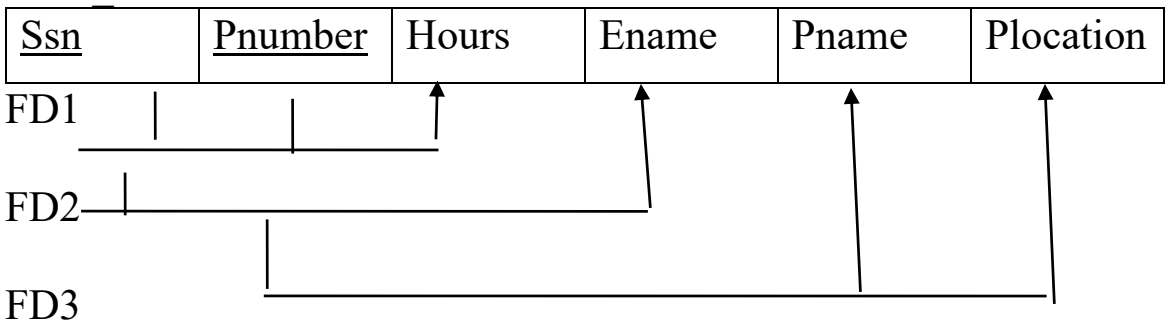
Area	County_name

LOTS1AX (Property_id#, Lot#, Area)
LOTS1AY (Area, County_name)

New FDs in DB are:
FD1: Property_id# → Lot#, Area
FD2: Area → County_name

4. Normalizing into 2NF and 3NF. (a) Normalizing EMP_PROJ into 2NF relations. (Done before but re-visited)

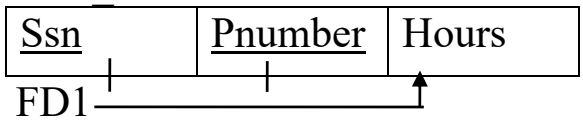
EMP_PROJ



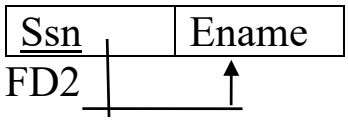
EMP_PROJ is in 1NF but not in 2NF because FD3 is violating the 2NF condition as it requires the primary key
Ssn, Pnumber ---FFD----> Pname
FD2 is also violating as well Ssn →Ename (violating)
FD3: Pnumber → Pname, Plocation (violating)

To make EMP_PROJ in 2NF, decompose it,
EMP_PROJ1 (SSn, Pnumber, Hours, ~~Ename~~, ~~Pname~~, ~~Plocation~~)
EMP_PROJ2 (Ssn, Ename)
EMP_PROJ3 (Pnumber, Pname, Plocation)

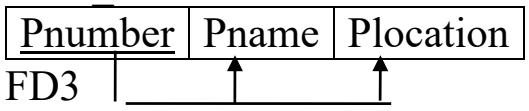
EMP_PROJ1



EMP_PROJ2



EMP_PROJ3



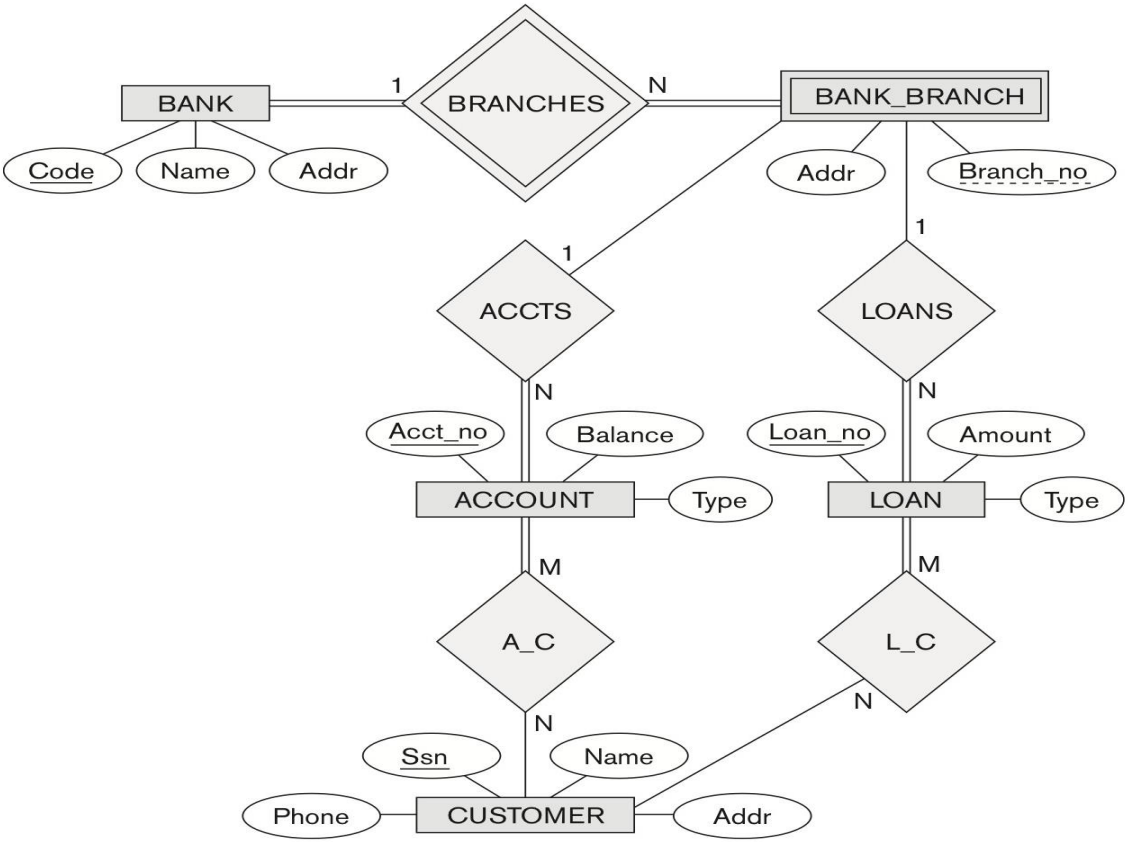
***

**** Parts of Solution to Midterm 1 Practice (handed out Fall 2020 Class)**

Section B (35 marks):
This section has 3 questions :

1. **(10 marks)** Given the ER diagram of the BANK database of Figure 3.22, answer the following questions with this diagram.

Figure 3.22 An ER diagram for a BANK database schema.



Write the relational database schema for BANK including all the entities with attributes, relationships with attributes and at least two referential integrity constraints that apply to this database.

COMPANY Database Component	COMPANY SCHEMA for the Database Component
Entities With their attributes (4 marks)	BANK(Code, Name, Addr) BANK_BRANCH(Branch_no, Addr) ---------- ACCOUNT(Acct_no, Balance, Type) LOAN(Loan_no, Amount, Type) CUSTOMER(Ssn, Name, Addr, Phone)
Relationships with their attributes (4 marks)	BRANCHES(Bank_Code, Branch_no) ACCTS(Acct_no, Branch_no) LOANS(Loan_no, Branch_no) A_C(Acct_no, Ssn) L_C(Loan_no , Ssn)
Two referential Integrity Constraints (2 marks)	The referential integrity constraints occur through foreign key attributes in a child relation (usually a relationship relation) that reference primary key attributes in a parent relation. Two examples are: 1. The foreign key attribute value for Bank_Code in the Branches relation must reference an existing Bank_Code value in the BANK entity relation. Also the foreign key attribute Branch_no value in the BRANCHES relation must reference an existing Branch_no in the BANK_BRANCH entity relation. 2. The foreign key attribute value for Bank_Code in the A_C relation must reference an existing Acct_no value in the ACCOUNT entity relation. Also the foreign key attribute SSn value in the A_C relation must reference an existing Ssn in the CUSTOMER entity relation

3. **(10 marks)** Design a simple database with at least 5 relations for such applications as “students taking courses taught by faculty in class rooms at a specific time”, “customers ordering items at specific dates and ordered items shipped from warehouse at a specific date”.

Student(sid, sname, major, gpa)
Course(cid, ctitle, credit)
Faculty(fid, fname, salary)
Room(Rid, location, capacity)

Enrolled(<u>sid</u>, <u>cid</u>, grade)
Teach(<u>fid</u>, <u>cid</u>, <u>Rid</u>, time)

Specify all the foreign keys in your designed database. State each foreign key using the following format "foreign key is the attribute ---------- of relation --------- that references relation ----------". (**5 marks**)

a. State two queries involving more than one table that can be posed on this database indicating the tables to visit to answer the queries each time. (**5 marks**)

Question	Foreign Key
2a (5 marks)	i. foreign key is the attribute ---sid---------- of relation ----Enrolled--------- that references relation --Student-------- ii. foreign key is the attribute ---cid--------- of relation -----Enrolled---- that references relation ---Course------ iii. foreign key is the attribute ---fid-------- of relation -----Teach------ that references relation --Faculty------- iv. foreign key is the attribute ---cid------- of relation ----Teach------ that references relation --course-------- v. foreign key is the attribute ----Rid------- of relation ----Teach-------- that references relation -Room-----

Question	Query	Tables needed to answer query
2b (5 marks)	i. Get the average gpa maintained by each student name Select Student.Sid, Student.Sname, avg(Student.gpa) From Enrolled, Student Where student.sid = Enrolled.sid Group by sid; (ii) Print all courses (titles) taken by student 'Smith' ~~Select Course.cid, Course.ctitle~~ ~~From Enrolled, Student, Course~~	i. Enrolled and Student ii. Enrolled, Student, Course

	~~Where Student.sid = Enrolled.sid And Course.cid = Enrolled.cid And Student.sname LIKE '%Smith';~~ .	

3. (15 marks)

a. Design a simple ~~normalized~~ Library database for tracking "customers reserve books" with not more than 3 tables and answer the following questions about your database. (**5 marks**)

~~b. Discuss how your database is in third normal form using functional dependencies (FDs).~~ (**4 marks**)

c. Create an instance of your database. (**4 marks**)

d. Provide one external level view query in English on your database. (**2 marks**)

Question	Answers
a. (5 marks)	CUSTOMER(cid, cname, caddr, cphone, cemail, number_of_books) BOOKS(bid, btitle, bcall_copies, no, npages, location) RESERVE(cid, bid, borrow_date, due_date)
b. (4 marks)	(not included in this test for this class) A database is in third normal form if all the tables in the database schema are in 3NF. A database table is in 3NF if every non-key attribute is functionally determined by only the primary key. This means that there is no transitivity in the functional dependency between an attribute of the table and the primary key. Looking at the 3 tables in the database schema above, RESERVE has primary key as: (cid, bid) with FDs as: (cid, bid) → , borrow_date, due_date. CUSTOMER has primary key as: (cid) with FDs as: (cid) → (cname, caddr, cphone, cemail, number_of_books). BOOKS has primary key as: (bid) with FDs as: (cid) → (btitle, bcall_no, npages, location)
c. (4 marks)	CUSTOMER ---------- cid cname caddr cphone email number_of_books 1 Mary Randolph 5199999991 mary 2 2 Peter California 5199999992 peter 1

	BOOKS -------- bid btitle bcall_no copies npages location 31 Intro to Databases CSDBW34 1 245 W3410 42 Intro to Java CSJVW33 2 300 W3305 RESERVE ------ cid bid borrow_date due_date 1 31 23-11-16 12-01-17 1 42 23-11-16 12-01-17 2 42 05-1216 30-01-17
d. (2 marks)	Show the customer, their email and title of all books that are due back.