**ISMG 6080 Homework 3 10 Points, Due Sunday 2/7, 11:00 pm. Submit a word document to Canvas.**

**10 Points. Graded on a 100-Point scale then scaled down using this formula**

**Hw3Grade=ScoreAsGraded/100\*10.**

**Problem 1 (5\*2 Point = 10 Points) Answer Yes or No** to the following Functional Dependency questions. Base your answers on the data shown in Table X below. There are no further rows in this table. **Provide reasoning VERY briefly.**

X:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F |
| a1 | b1 | c1 | d1 | 13 | f1 |
| a1 | b1 | c2 |  | 13 | f1 |
| a1 | b2 | c1 | d1 | 14 | f1 |
| a3 | b2 | c3 |  | 14 | f1 |
| a3 | b3 | c3 | d1 | 14 | f1 |
| a5 | b3 | c2 |  | 14 | f1 |

1). Does A→C? **No,** a1 in column A has multiple values in column C, a1 corresponds to c1 & c2.

2) Does B→F? **Yes**, each value in column B is corresponds to a single value in F column.

3). Does B+C→D? **Yes,** combination of B + C has unique value in column D and there are no multiple values for the same combination.

4). Does C+E→A **Yes,** Combination of values from column C + E is unique, and every combination has a unique value in column A

5). Does A+C→D? **No**, Combination of a3 & c3 has two values in Column D – d1 & null.

Hint: consider null value as a regular value.

**Problem 2 (10 Points)**

The following three tables capture part of the information about a small airline operation. On each flight, individual seats are not assigned.

**ScheduledFlightInfo**:(Flightno, from, to, departureTime, arrivalTime)

**ActualFlightInfo**: (InfoId, FlightDate, FlightNo, actualDepartureTime, ActualArrivalTime, employeeNo, role, seats-remaining, airplane-serialNo)

**Employee**: (EmployeeNo, name, phone, jobTitle)

**Identify functional dependency (or dependencies) for each numbered paragraph below**. Use only fields in a table listed above. Consider all information given together.

**Identify which table each functional dependency is defined on**. Do **NOT** add columns to the above two tables.

***Example:***

* Each employee has a unique employeeNo. Database records each employee’s current name, phone and job title, such as pilot, flight attendant, and machinist

*Answer: Employee Table, FD:* EmployeeNo🡪name, phone, jobTitle

1. Each flight number refers to ONE nonstop daily flight that goes between “from” city and “to” city. The table records its scheduled departure time and arrival time. Flight numbers are unique.

**Table**: **ScheduledFlightInfo**

**FD:** Flightno 🡪 from, to, departureTime, arrivalTime

1. The actual departure and arrival time of a flight on a given day are recorded in the ActualFlightInfo table. A given flight number (say U556) is only flown once on any given day. FightDate is the date of departure.

**Table: ActualFlightInfo**

**FD:** FlightDate + FlightNo 🡪 actualDepartureTime, ActualArrivalTime

1. On any actual flight (Say U556 on 2/10/2016), a crew serves on it. The database records the employeeNo and the role (e.g., captain, 1st flight attendant) for each crewmember on each actual flight scheduled or flown on a particular date (e.g., Adam served as the captain on U556 on 2/10/2016). Each employee serves only one role on a given flight scheduled or flown. An employee may serve different roles on different flights. For example, Adam served as the caption of U556 on 2/10/2016 but he served as the co-pilot on U556 on 2/13/2016. Mike served as the first flight attendant on U556 on 2/10/2016 but he served as the second flight attendant on U556 on 2/13/2016. (Hint: The FD should look like? 🡪 Role)

**Table: ActualFlightInfo**

**FD:** FlightDate + FlightNo + employeeNo 🡪 Role

1. On any actual flight (Say U556 on 2/10/2016), the plane flying the flight is recorded in the database. Obviously, there is only one plane flying for a given flight scheduled or flown, say U556 on 2/10/2016. However, U556 on 2/13/2016 can be flown with a different airplane. Airplane-seriesNo is unique to each airplane.

**Table: ActualFlightInfo**

**FD:** FlightDate + FlightNo 🡪 airplane-serialNo

1. Number of remaining seats on each actual flight (say U556 on 9/20/2016) is recorded in the database.

**Table: ActualFlightInfo**

**FD:** FlightDate + FlightNo 🡪 seats-remaining

**Problem 3 (2\* 5 Points = 10 Points) Normalize to 1NF**

**A:** Normalize the following table into first normal form. The table uses one row to record information about each student. A student may take one or more electives. This table is not in 1NF. Normalize this table so it is in 1NF. In your answer,

* **list** all columns of the new table,
* **identify** the primary key, and
* **enter** all data for the normalized table.

|  |  |  |  |
| --- | --- | --- | --- |
| **SID** | **LName** | **FName** | **Electives** |
| 100345 | Ford | George | Geometry, Dance |
| 200898 | Gibbs | Mary | Orchestra |
| 300987 | Jordan | Jeff | Orchestra, Creative Writing, Dance |

Solution:

**PK** – SID + Electives

|  |  |  |  |
| --- | --- | --- | --- |
| **SID** | **LName** | **FName** | **Electives** |
| 100345 | Ford | George | Geometry |
| 100345 | Ford | George | Dance |
| 200898 | Gibbs | Mary | Orchestra |
| 300987 | Jordan | Jeff | Orchestra |
| 300987 | Jordan | Jeff | Creative Writing |
| 300987 | Jordan | Jeff | Dance |

**B:** Normalize the following table into first normal form. The table uses one row to record information about each purchase transaction. Each transaction has a unique ID assigned to it. The transaction completion date/time is recorded (date/time when receipt is printed). A customer may purchase multiple items in one purchase transaction. A customer may purchase multiple units of the same item as well. This table is not in 1NF. Normalize this table so it is in **1NF**. In your answer

* **list all columns of the new table,**
* **identify** the primary key, and
* **enter** all data for the normalized table.

|  |  |  |  |
| --- | --- | --- | --- |
| **TranID** | **TDateTime** | **Items** | **Quantities** |
| 20 | 1/3/2015 3:07 pm | SKU123, SKU45, SKU89 | 1, 3, 1 |
| 21 | 1/4/2015 4:15 pm | SKU123 | 2 |
| 22 | 1/5/2015 9:08 am | SKU123, SKU88, SKU9 | 1,1, 3 |

**PK –** TranID + Items

|  |  |  |  |
| --- | --- | --- | --- |
| **TranID** | **TDateTime** | **Items** | **Quantities** |
| 20 | 1/3/2015 3:07 pm | SKU123 | 1 |
| 20 | 1/3/2015 3:07 pm | SKU45 | 3 |
| 20 | 1/3/2015 3:07 pm | SKU89 | 1 |
| 21 | 1/4/2015 4:15 pm | SKU123 | 2 |
| 22 | 1/5/2015 9:08 am | SKU123 | 1 |
| 22 | 1/5/2015 9:08 am | SKU88 | 1 |
| 22 | 1/5/2015 9:08 am | SKU9 | 3 |

**Problem 4 (2\*5 Points= 10 Points) Given functional dependencies, identify the primary key.**

**A:** Table "Complaints" records complaints related information. Given the functional dependencies identified below, **identify the primary key**. **Provide the process through which you derived the primary key.** Do not assume any further functional dependencies.

**Complaint:** (Complaint#, Transaction#, TransactionDate, CustID, CustName, CustAddress, ReviewDept)

All Functional Dependencies:

Complaint# 🡪 Transaction#, CustId

Transaction#🡪TransactionDate

CustID 🡪 CustName, CustAddress

**PK =** Complaint# + ReviewDept

**Derivation Process:**

**Step 1: Adding LHS 🡪** Complaint# + Transaction# + CustID

**Step 2:** reducing by removing the column Transaction# and CustId since it appears in RHS 🡪 Complaint#

**Step 3:** Adding columns which is not in both LHS & RHS 🡪 Complaint# + ReviewDept

**B:**  Table "Donation" records donations received and related information. Given the functional dependencies identified below, **identify the primary key**. **Provide the process through which you derived the primary key.** Do not assume any further functional dependencies.

**Donation:** (Donation#, RepId, RepName, DonorId, DonorName, DonorPhone, Event)

All Functional Dependencies:

Donation# 🡪RepId, Event, DonorId

DonorID 🡪 DonorName, DonorPhone

RepId🡪RepName

**PK =** Donation#

**Derivation Process:**

**Step 1: Adding LHS 🡪** Donation # + DonorID + RepId

**Step 2:** reducing by removing the column DonorID and RepId since it appears in RHS 🡪 Donation #

**Step 3:** Adding columns which is not in both LHS & RHS, in this table there is nothing to add 🡪 Donation #

**Problem 5 (5\* 4 Points = 20 Points)**. Normalize each table to 2NF given the primary key (underlined in each table) and additional functional dependencies.

For each table below, determine if the table satisfies 2NF. If the table is already in 2NF, **say so**. If a table violates 2NF, **say so** and produce the 2NF tables. **It is sufficient to just list the results of 2NF normalization. Make sure to indicate the primary key of each table you write by underlining it.**

1. Book: (BookTitle, DateofPrint, #copiesPrinted, #copiesSold, Price, MajorSponsorName, MajorSponsorPhone )

Additional FD: MajorSponsorName → MajorSponsorPhone

Table already satisfy 2NF

B. ProjectMileStone: (ProjId, MilestoneId, MilestoneDesc, CompletionDate)

Additional FDs: None

Since there is no functional dependency in the given table, final table will be (ProjId, MilestoneId, MilestoneDesc, CompletionDate)

C. Tutoring: (TutorId, StudentID, TutorFName, TutorLName, Sfname, SLName, #ofTimes, TotalFee)

Additional FD: TutorId#→TutorFName, TutorLName

StudentId🡪SFName, SLName

T1: (StudentId, SFName, SLName)

T2: (TutorId#, TutorFName, TutorLName)

T3: (TutorId, StudentID, #ofTimes, TotalFee)

1. FileDownload: (SessionId, DownLoadId, FileName, SessionStDateTime, SessionEndDateTime, FileSize, FileType)

Additional FDs: SessionId → SessionStDateTime, SessionEndDateTime

FileName 🡪 FileSize, FileType

**T1:** SessionId, SessionStDateTime, SessionEndDateTime

**T2:** SessionId, DownLoadId, FileName, FileSize, FileType

1. TaskForce: (TaskforceId, MemberId, MLName, MFName, TFName, LeaderId, LLName,LFName, MemStart, MemEnd)

Additional FDs: TaskForceId 🡪 TFName, LeaderId

LeaderId 🡪 LLName, LFName

MemberId 🡪 MLName, MFName

T1: (TaskForceId, TFName, LeaderId, LLName, LFName)

T2: (MemberId, MLName, MFName)

T3:(TaskforceId, MemberId, MemStart, MemEnd)

**Problem 6 (6 Points):** Refer to Problem 6 on Ex-3. Note that I’ve changed the data in the table.

**EmployeePrj**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EmployeeId | LastName | FirstName | ProjectId | Assignment  StartDate | Assignment  EndDate |
| 1001 | Park | Amy | Proj1 | 6/4/2016 | 7/8/2016 |
| 1001 | Park | Amy | Proj2 | 10/11/2016 | 11/9/2016 |
| 1002 | Walters | Mary | Proj2 | 10/11/2016 | 11/9/2016 |
| 1002 | Walters | Mary | Proj3 | 1/5/2017 |  |
| 1003 | Smith | Carol | Proj3 | 1/7/2017 |  |
| 1004 | Gupta | Alok | Proj1 | 6/3/2016 | 7/8/2016 |
| 1004 | Gupta | Alok | Proj2 | 10/1/2016 | 11/9/2016 |
| 1004 | Gupta | Alok | Proj3 | 1/8/2017 |  |

Given the business rules as given in Ex3, Problem 6, it is not in 2NF. Normalize it will give the following two tables (see the solution file for details):

Employee: (EmployeeID, LastName, FirstName)

ProjectAssignment: (EmloyeeId, ProjectId, StartDate, EndDate)

**For this homework problem, present the final result of the 2NF tables (i.e., Employee table and ProjectAssignment table above) with data in them using data seen here.**

**Employee:**

|  |  |  |
| --- | --- | --- |
| EmployeeId | LastName | FirstName |
| 1001 | Park | Amy |
| 1002 | Walters | Mary |
| 1003 | Smith | Carol |
| 1004 | Gupta | Alok |

**ProjectAssignment:**

|  |  |  |  |
| --- | --- | --- | --- |
| EmployeeId | ProjectId | Assignment  StartDate | Assignment  EndDate |
| 1001 | Proj1 | 6/4/2016 | 7/8/2016 |
| 1001 | Proj2 | 10/11/2016 | 11/9/2016 |
| 1002 | Proj2 | 10/11/2016 | 11/9/2016 |
| 1002 | Proj3 | 1/5/2017 |  |
| 1003 | Proj3 | 1/7/2017 |  |
| 1004 | Proj1 | 6/3/2016 | 7/8/2016 |
| 1004 | Proj2 | 10/1/2016 | 11/9/2016 |
| 1004 | Proj3 | 1/8/2017 |  |

**Problem 7 (10 Points)**

**For the table below, perform the following 3 steps. The primary key of each table is underlined. Additional functional dependencies are given. Do not assume further functional dependencies.**

**Step 1:** Determine if the table satisfies 2NF. If the table is already in 2NF, **say so**. If a table violates 2NF, **say so, give the reasoning,** and produce the 2NF tables.

**Step 2: D**etermine if all resulting tables from Step 1 satisfy BCNF. For each of the resulting tables from Step 1, if it is already in BCNF, say so.If it violates BCNF, **say so, give the reasoning,** and produce the BCNF tables.

**Step 3**: Check and make sure every table you wrote in Steps 1 and 2 has a primary key specified. If not, specify the primary key for each table missing the primary key.

**Indicate the primary key of each table that you write by underlining it. Points will be deducted if a table you write does not have the PK underlined.**

**Turn Off Spelling Checker so you can see the primary key for each table.**

EventBooth: (EventId, BoothNum, Loc, EventTotal, BoothRevenue, LocPhone)

Additional FDs: EventId → EventTotal

Loc → LocPhone

**Step 1:** EventId 🡪 EventTotal, Since EventId id the primary key it violates the 2NF, to normalise it we reduce the table into two as follows.

**T1: (**EventId, EventTotal)

**T2: (**EventId, BoothNum, Loc, BoothRevenue, LocPhone)

**Step 2:** T2 has the functional dependency Loc 🡪 LocPhone, so it violates the BCNF, we will normalize as follows.

**T1: (**EventId, EventTotal)

**T2: (**EventId, BoothNum, Loc, BoothRevenue)

**T3: (**Loc, LocPhone)

**Step3:** T1 and T2 has the primary key, but newly created table has no primary key.

Primary keys are:

T1: EventId

T2: EventId, BoothNum

T3: Loc

**5**

**Problem 8** (**2\* 7 Points Each = 14 Points**)

Determine if each of the following tables satisfy BCNF. If the table is already in BCNF, **say so**. If the table violates BCNF, **say so, give the reasoning,** and produce the BCNF tables. Do not go normalize it to 2NF first. Normalize each table to BCNF directly. The primary key (underlined in each table) and additional functional dependencies are already given. Do not assume further function dependencies. **Be sure to identify the primary key for each table you write by underlining it.**

1. TaskForce: (TaskforceId, MemberId, MLName, MFName, TFName, LeaderId, LLName,LFName, MemStart, MemEnd)

Additional FDs: TaskForceId 🡪 TFName, LeaderId

LeaderId 🡪 LLName, LFName

MemberId 🡪 MLName, MFName

Table is not in 2NF and it has functional dependency, this violates the BCNF.

**Normalized table are:**

**T1:** (TaskForceId, TFName, LeaderId)

**T2:** (LeaderId, LLName, LFName)

**T3:** (MemberId, MLName, MFName)

**T4:** (TaskforceId, MemberId, LeaderId, MemStart, MemEnd)

1. FileDownload: (SessionId, DownLoadId, FileName, SessionStDateTime, SessionEndDateTime, FileSize, FileType)

Additional FDs: SessionId → SessionStDateTime, SessionEndDateTime

FileName 🡪 FileSize, FileType

Table is not in 2NF and it has functional dependency, this violates the BCNF.

**Normalized table are:**

**T1:** SessionId, SessionStDateTime, SessionEndDateTime

**T2:** SessionId, DownLoadId, FileName

**T3:** FileName, FileSize, FileType

**Problem 9: (10 Points).** The following table records performance gigs.

Performance: (GigId, GigName, GigDate, StartTime, EndTime, LocationId, LName, BandId, BandName)

Additional FDs: BandId🡪BandName

LocationId🡪LName

| **GigID** | **GigDate** | **Gig Name** | **Start TIme** | **End Time** | **LocationID** | **Lname** | **BandID** | **Band Name** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 12/28/2015 | JL Brunch | 10:30 AM | 12:30 PM | 1 | Juciy Lucy's Patio | 1 | Acoustic Mayhem |
| 2 | 1 /9 /2015 | Fins Bar | 09:00 PM | 11:00 PM | 11 | Sams Lodge | 2 | Leonard Curry Tri |
| 3 | 1 /10/2015 | Private Birthday | 09:00 PM | 11:00 PM | 1 | Juciy Lucy's Patio | 1 | Acoustic Mayhem |
| 4 | 1 /11/2015 | Ski Spree | 10:00 AM | 12:00 PM | 11 | Sams Lodge | 1 | Acoustic Mayhem |
| 5 | 1 /15/2015 | Lime Light 5 year Anniversary | 10:00 PM | 11:30 PM | 9 | Lime Light | 1 | Acoustic Mayhem |
| 6 | 1 /16/2015 | Aspen Ski spree event party | 12:00 PM | 01:30 AM | 10 | Fannys Lodge | 2 | Leonard Curry Tri |

Given the primary key and additional functional dependency above, answer the following questions:

1. Is this table in 2NF? If not, normalize it to 2NF.

Yes, Table already satisfy 2NF.

1. Are the resulting tables in BCNF? If not, normalize them to BCNF.

Yes, it violates the BCNF, since there is a functional depency on the table.

T1: (GigId, GigName, GigDate, StartTime, EndTime, LocationId, BandId)

T2: (LocationId, LName)

T3: (BandId, BandName)

1. Show final BCNF tables with data in them.

| **GigID** | **GigDate** | **Gig Name** | **Start TIme** | **End Time** | **LocationID** | **BandID** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 12/28/2015 | JL Brunch | 10:30 AM | 12:30 PM | 1 | 1 |
| 2 | 1 /9 /2015 | Fins Bar | 09:00 PM | 11:00 PM | 11 | 2 |
| 3 | 1 /10/2015 | Private Birthday | 09:00 PM | 11:00 PM | 1 | 1 |
| 4 | 1 /11/2015 | Ski Spree | 10:00 AM | 12:00 PM | 11 | 1 |
| 5 | 1 /15/2015 | Lime Light 5 year Anniversary | 10:00 PM | 11:30 PM | 9 | 1 |
| 6 | 1 /16/2015 | Aspen Ski spree event party | 12:00 PM | 01:30 AM | 10 | 2 |

| **LocationID** | **Lname** |
| --- | --- |
| 1 | Juciy Lucy's Patio |
| 11 | Sams Lodge |
| 9 | Lime Light |
| 10 | Fannys Lodge |

| **BandID** | **Band Name** |
| --- | --- |
| 1 | Acoustic Mayhem |
| 2 | Leonard Curry Tri |