RMIT UNIVERSITY

School of Science

# ISYS2095 Database Concepts

Assessment 3: Database Design

Assessment Type: PDF

Word limit: N/A (see instructions)

Due date: Monday 22nd February 2021 23:59 (AEDT)

20% of your overall grade

## Overview

The objective of this assignment is to measure your understanding of functional dependencies, normalisation and their application to database design, and SQL triggers.

## Assessment Criteria

This assessment will measure your ability to:

- identify issues with and compare and justify relational database design using functional dependencies.
- create SQL triggers

## Course Learning Outcomes

This assessment is relevant to the following Course Learning Outcomes:

- CLO1: Describe the underlying theoretical basis of the relational database model and apply the theories into practice.
- CLO 2: Explain the main concepts for data modeling and characteristics of database systems.
- CLO 3: Identify issues with, compare and justify relational database designs using functional dependency concepts.

- CLO 4: Develop a database based on a sound database design.

# Assessment details

## Task 1 (20 marks)

In a database about competitors in the 2021 Tokyo Olympics, information held includes *CID* and *Cname* (the ID and name of the competing sportsperson), *TeamName*, *TeamCoach*, *EventName*, *Event-Date* and *TeamRating*.

Assume the following functional dependencies apply:

CID -> Cname, TeamName
TeamName -> TeamCoach
EventName -> EventDate
TeamName, EventName -> TeamRating

Consider an initial design of a relation schema for the database:

```
Competitor(CID, EventName, Cname, TeamName, TeamCoach, EventDate, TeamRating)
```

Q1.1

- Indicate the primary key of the relation.
- Identify and describe any sources of redundancy.
- Illustrate, using one example for each type of anomaly, how the Competitor relation could be subject to an
  - insert anomaly
  - deletion anomaly
  - update anomaly

Q1.2

- Perform a lossless decomposition of the Competitor relation into relations in 3NF. For each resulting relation, underline the primary key and denote any foreign keys with asterisks (*).

## Task 2 (24 marks)

'HiTech Computer Security' performs computer security audits for contracted customers. Each customer has a unique customer number and an address (more than one customer may be at any given address). When a HiTech engineer has to perform audits they are allocated a van to use on that day. If the engineer finishes auditing all their customers for that day, the van could be reallocated to another engineer. Each van has a unique van number and each engineer has a unique staff number. On any given day, HiTech will only audit a given customer once.

An initial schema for HiTech is:

```
Audit(custAddress, custNum, date, staffNum, staffName, vanNum)
```

Q2.1

- Identify and describe two sources of redundancy in the Audit relation.
- Illustrate, using examples, how the Audit relation could be subject to an
  - insert anomaly
  - deletion anomaly
  - update anomaly

Q2.2

- List the functional dependencies. Do not include trivial functional dependencies.
- Indicate possible candidate key(s) of the Audit relation.

Q2.3

- Explain, with the assistance of functional dependencies, the normal form of the Audit relation.

Q2.4

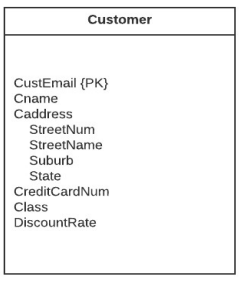
- Perform a lossless decomposition of the Audit relation into relations in 3NF. For each resulting relation, underline the primary key and denote any foreign keys with asterisks (*).

## **Task 3** (29 marks)

Consider the following business rules for an online store.

- Customers register their name, address (consisting of street number, street name, suburb, state), credit card number and email account with the store, which uses their email account as a key. The store also records the customer's class (one of Gold, Silver or Bronze, with the initial value of Bronze) and the customer's discount rate (10% for Gold class, 5% for Silver class and 0% for Bronze class customers).
- A customer can make online orders, each of which has an order id, an order date, a shipping date and the products and their quantities making up the order. Products have a product code, a product name, a description, one or more technical specifications and a unit price.

A *customer* entity has already been modelled from the above rules, as follows:




Q3.1

Based on the above information, and utilising the above *customer* entity as is, complete the database requirements as an Entity-Relationship (ER) diagram. Carefully state any assumptions that you make. In your ER diagram, you must properly denote all applicable concepts, including weak or strong entities, keys, composite or multi-valued attributes; relationships and their participation and cardinality constraints.

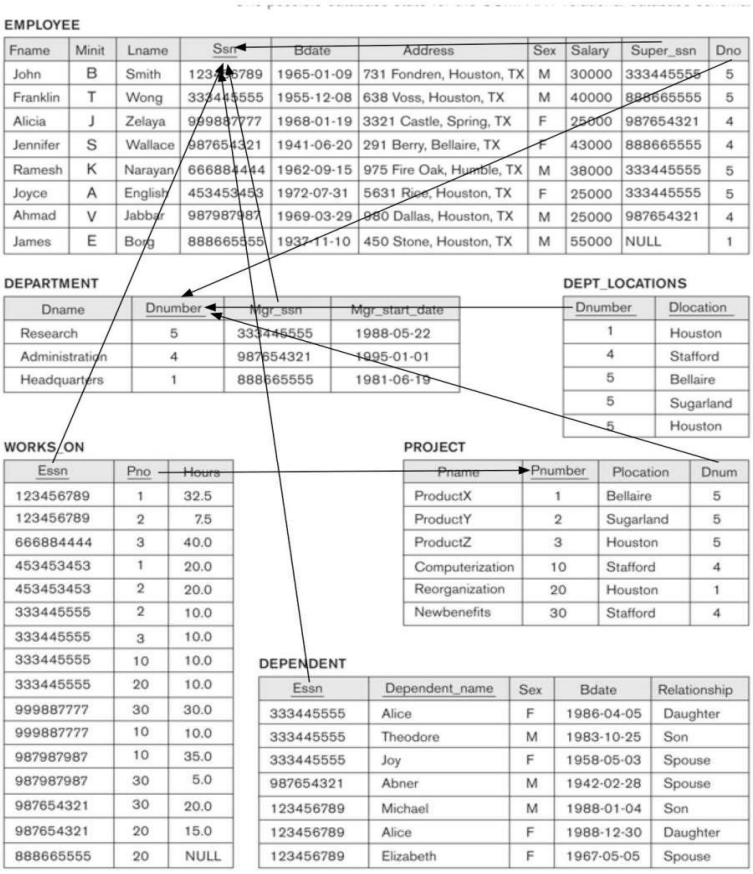
Q3.2
Map the ER diagram from Q3.1 into a final relational database schema using the 4 step process described in week 4. Show every step of the mapping. No marks are awarded to the final schema if you do not show the partially built schema at the end of each step. Indicate the primary key (underlined) and any foreign key (with an asterisk) in each relation.

Q3.3
Validate each of the relations in the model using the normalisation process. For each relation:

- Give the (non-trivial) functional dependencies.
- For each normal form (1NF, 2NF, 3NF) state if the relation is that form. Justify your answer.
- If a relation is not in 3NF, perform a lossless decomposition of it to 3NF.

## **Task 4** (7 marks)

This task uses a relational database schema and instance adapted from Fundamentals of Database Systems, Elmasri and Navathe. (Question 5.11), given below and available as an SQLite database (Company.db) on the course Canvas shell under module Resources: Sample Databases and Tools.



**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 |

**WORKS_ON**

| Essn | Pno | 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 |

**DEPENDENT**

| Essn | Dependent_name | Sex | Bdate | Relationship |
|---|---|---|---|---|
| 333445555 | Alice | F | 1986-04-05 | Daughter |
| 333445555 | Theodore | M | 1983-10-25 | Son |
| 333445555 | Joy | F | 1958-05-03 | Spouse |
| 987654321 | Abner | M | 1942-02-28 | Spouse |
| 123456789 | Michael | M | 1988-01-04 | Son |
| 123456789 | Alice | F | 1988-12-30 | Daughter |
| 123456789 | Elizabeth | F | 1967-05-05 | Spouse |



Most of the attribute names are self-explanatory. Super_SSN refers to corresponding employee's supervisor's SSN (Social Security Number). This example is based on US system, assume SSN is similar to Australian Tax File Number.

Arrows indicate foreign keys and the corresponding attributes in parent relation. In the case of Super_SSN, the parent relation is the Employee relation itself (self-referencing).

Assume the following table is added to the Company database:

```
CREATE TABLE works_on_total (
      essn NUMERIC (9) NOT NULL,
      totalProjects INTEGER,
      totalHours NUMERIC (9, 2),
      PRIMARY KEY (essn),
      FOREIGN KEY (essn) REFERENCES employee (ssn)
);
```

Assume table works_on_total already has a row for each employee, detailing the current total number of projects they are working on, and the total number of hours that they are working.

Q4.1
Write an SQL AFTER trigger for updating an employee's works_on_total entry each time they have a project added to table works_on to update the current total number of projects they are working on, and the total number of hours that they are working.

## Referencing guidelines

Use Harvard referencing style for this assessment. You must acknowledge all the courses of information you have used in your assessments.

Refer to the RMIT Easy Cite referencing tool to see examples and tips on how to reference in the appropriated style. You can also refer to the library referencing page for more tools such as EndNote, referencing tutorials and referencing guides for printing.

## Submission

You should submit one PDF file with all answers together. Please preface each answer with a comment indicating the question (e.g: --Task 1.1). Submit to the assessment page in canvas by the due date.

## Academic integrity and plagiarism

Academic integrity is about honest presentation of your academic work. It means acknowledging the work All code or other material that is not original must be fully credited. That is, any material that is copied or derived from another source must be clearly identified as such and the original author must be identified. Sometimes students assist each other with an assignment, but end up working together too closely, so that the students' separate solutions have significant parts in common; unless the solutions were developed independently, they are regarded as plagiarised.

Plagiarism is a very serious offence. Any submissions determined to be a result of plagiarism will be deemed as an academic misconduct and harsh penalties apply. It is also an offence for students to allow their work to be plagiarised by another student. You should familiarize yourself with the university website for Academic Integrity

Policy, Procedures and Guidelines: https://www.rmit.edu.au/students/student-essentials/rights-andresponsibilities/academic-integrity

All work is to be done individually and plagiarism of any form will be dealt with according to the RMIT plagiarism policy.

## Penalties for late submissions

Late submissions of assignments will be penalised as follows. For 1 to 5 days late, a penalty of 10% (i.e. 10% out of total marks, not 10% out of your marks) per day. For assignments more than 5 days late, 100% penalty applies.

## Special Consideration

If unexpected circumstances affect your ability to complete the assignment you can apply for special consideration. If you seek a short extension, you can directly contact the lecturer. For longer extensions, you must follow instructions provided at: http://www1.rmit.edu.au/students/specialconsideration

## Assessment declaration

When you submit work electronically, you agree to the assessment declaration.