School of Science

ISYS2095 Database Concepts

Assessment 1: Database Design

Assessment type: PDF Word limit: N/A

Due Date: 3 July at 23:59 (Melbourne time)

Weighting: 20%, 20 Marks

Overview

The objective of this assignment is to measure your understanding of the basic concepts in the relational database model and using entity-relationship model for database design. The assessment is in two parts, split into four tasks which cover Basic ER Modelling and Basic Relational Modelling. The tasks are as follows.

Part A: Entity-Relationship Modelling (12 Marks)

1. Design and plan for the implementation of a database system, diagramming the design to a high standard using UML notation through the diagramming tool Lucidchart.

2. Model the activities of an organisation and present the model as an Entity-Relationship (ER) diagram. Analyse this ER diagram, and possibly modify it, based on additional client requirements.

3. Map an ER diagram into a relational database schema, showing every step of the mapping.

Part B: Relational Database Model (8 Marks)

4. Answer a series of short questions about a Relational Database model.

To complete this assessment, you must be familiar with Lucidchart, which is covered during the Week 1-4 activities.

Assessment criteria

This assessment will measure your ability to:

- Describe various data modelling and database system technologies.
- Explain the main concepts for data modelling and characteristics of database systems.
- Identify issues with, compare and justify relational database designs using the functional dependency concepts.

Course learning outcomes

This assessment is relevant to the following course learning outcomes:

CLO1	Describe various data modelling and database system technologies.
CLO2	Explain the main concepts for data modelling and characteristics of database systems.
CLO3	Identify issues with, compare and justify relational database designs using the functional dependency concepts.

Assessment details

Part A: Entity-Relationship Modelling

Task 1: Designing an Entity-Relationship Model

The House Transact Mate (HTM) Case Study

House Transact Mate (HTM) is a real-estate agency that has offices throughout Melbourne. The following are the requirements for managing data about staff, offices, and properties for HTM.

Offices

HTM has branch offices throughout Melbourne. Each branch office is allocated several staff members, including a manager who manages the operations of the office. The data describing a branch office includes a unique branch number, address (i.e., street, suburb, and postcode), up to 3 telephone numbers, and the name of the staff member who currently manages the office. Additional data about managers includes the date that the manager started the position at the current branch office and a monthly bonus payment based on performance.

Staff

Staff members with the role of supervisor are responsible for the day-to-day activities of a team of staff called agents. Each supervisor is responsible for up to a maximum of 10 agents at any one time. Not all staff members are assigned to a supervisor. The data for each staff member includes a unique staff number, name, address, position, salary, name of supervisor (where applicable), and the details of the branch office where a staff member is working.

Properties

Each branch office offers a range of properties for rent. The data stored for each property includes a unique property number, address (i.e., street, suburb, postcode), type, number of rooms, and weekly rent. The management of a property is assigned to an agent, and an agent manages a maximum of 100 properties at any one time. When a property is available for rent, the property details will be displayed on the HTM website and, when necessary, as advertisement on other websites.

Based on the given description, model the business rules of HTM, and present your model 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 cardinality and participation constraints.

If you cannot represent any of this information in the ER model, clearly explain what limitations in the ER model restrict you from representing your model.

You **must** use UML notation and the diagramming tool Lucidchart to draw your diagram. Your diagram must be drawn to a high standard with minimal clutter. You are **not** required to map the ER model to relational model.

A special note: This is an open-ended question with many different models that can be derived. Your model is assessed based on how accurately it represents business rules described above.

Task 2: Designing an Entity-Relationship Model

Attention: Correction in v2.1

A typo is corrected in item 2 of the case study below. The correction is highlighted.

Part A: Initial Design

Deliveraa Case Study

Deliveraa is a logistics and transport company. You are asked to design a database for scheduling their transport jobs. Requirements for the database are as follows:

1. Each truck has a unique registration number. It also has a type and the year it is made.
2. Each customer has a unique ABN. It also has a name, and several contact information (e.g., mobile, landline, fax, email etc). There are no duplicates in contact information, that is, for example, the system must ~~now~~ not allow the same email address to be recorded twice.
3. Delivery jobs are identified by a unique job number. They also have “from” and “to” destinations, and estimated distance to travel.
4. Drivers are identified by a unique employee ID. They also have first name, last name, address, and phone number.
5. A customer may book several delivery jobs. A single job is for only one customer who “owns” the job.
6. A driver can be assigned several delivery jobs, but a job has only one driver. Some description about the job is also recorded. Each job has a Status (e.g., open, complete, etc).
7. A job is assigned only one truck, but a truck can be assigned to several jobs.

Based on the given description, model the business rules of Deliveraa, and present your model 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 cardinality and participation constraints.

If you cannot represent any of this information in the ER model, clearly explain what limitations in the ER model restrict you from representing your model.

You **must** use UML notation and the diagramming tool Lucidchart to draw your diagram. Your diagram must be drawn to a high standard with minimal clutter. You are **not** required to map the ER model to relational model.

A special note: This is an open-ended question with many different models that can be derived. Your model is assessed based on how accurately it represents business rules described above.

Part B: Client Adjustments

After presenting your ER model to Deliveraa management, you are asked if it can be used to perform the following additional task.

8. Each driver can accept or reject a job that is assigned to him/her. The management wants to be able to produce a report showing the history of all the jobs that a particular driver refused to accept.

9. When customers use the services of Deliveraa for some time, they might wish to nominate certain drivers as their favourite drivers.

Explain, referring to your ER model, how to produce the requested list. If it is not possible to produce the list from the model, state why, modify the model, provide the modified ER diagram (in addition to your original ER diagram), and explain how it can accommodate this requirement.

Task 3: Mapping an ER Model to a Relational Database Schema

Attention: Correction in v2

The ER diagram in Figure 1 is slightly updated to include an assumption, which must be considered when completing the question.

Consider the following ER diagram, which shows aspects of performances of shows on cruise ships by various performers.

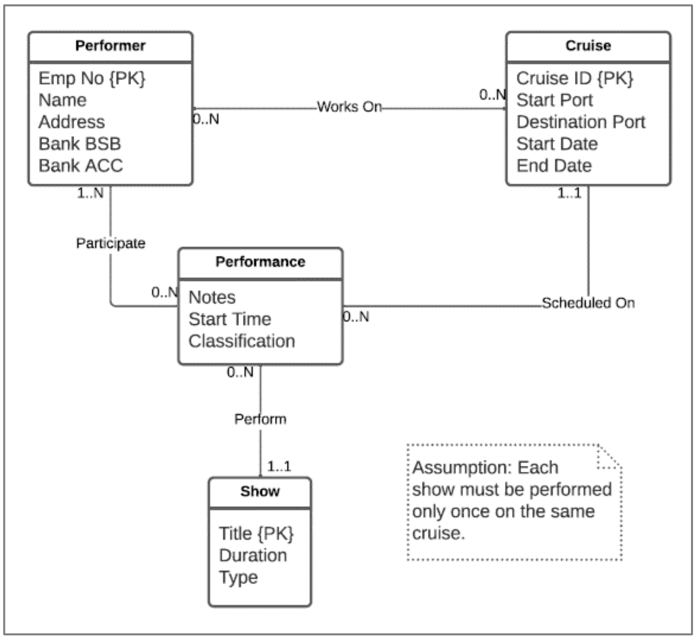


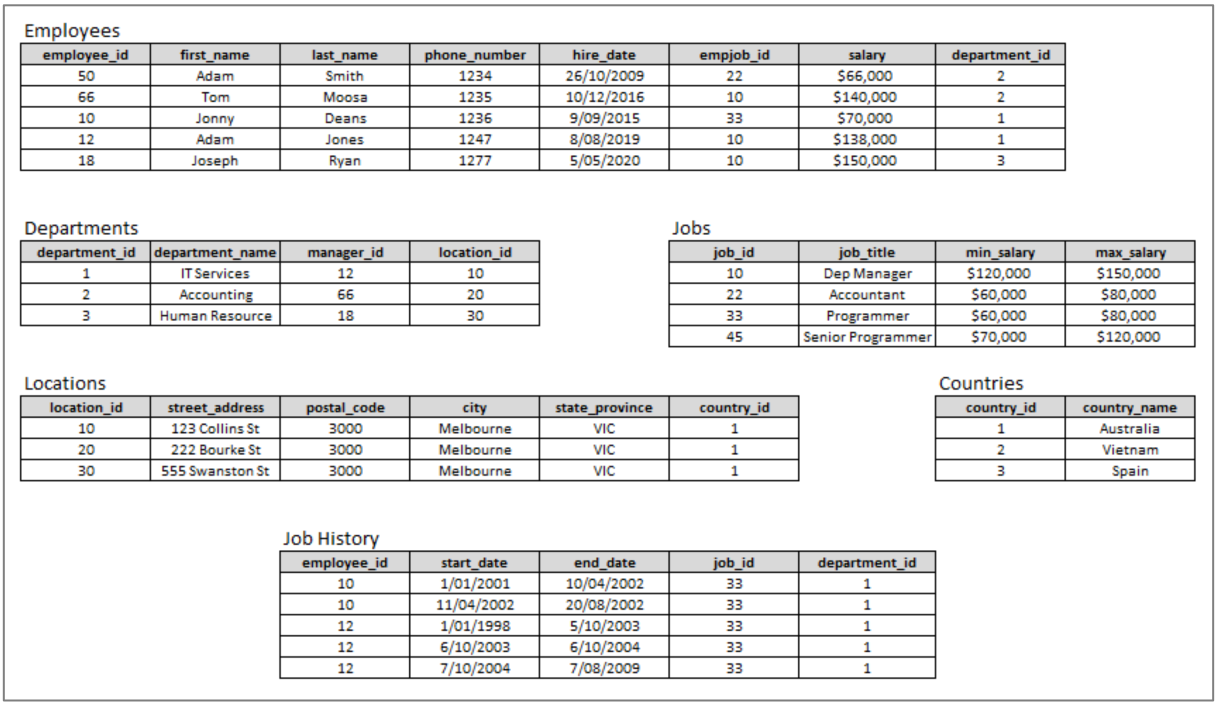
Figure 1: Cruise Performance ER Diagram

You are requested to map the above ER diagram into a relational database schema. 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. Clearly indicate the primary key (underlined) and foreign keys (with an asterisk) in each relation.

Part B

Task 4: Relational Database Model

This section contains the schema and a database instance for the Employee database that stores employee data for an organisation. The data includes items such as personal info (e.g., name, phone, salary), departments of the organisation (e.g., name and location of each department, who the manager is), jobs (e.g., titles, salary range), and a history for past contracts with each employee. A database instance is shown in Figure 2 followed by the database schema.



Employees

employee_id	first_name	last_name	phone_number	hire_date	empjob_id	salary	department_id
50	Adam	Smith	1234	26/10/2009	22	$66,000	2
66	Tom	Moosa	1235	10/12/2016	10	$140,000	2
10	Jonny	Deans	1236	9/09/2015	33	$70,000	1
12	Adam	Jones	1247	8/08/2019	10	$138,000	1
18	Joseph	Ryan	1277	5/05/2020	10	$150,000	3

Departments

department_id	department_name	manager_id	location_id
1	IT Services	12	10
2	Accounting	66	20
3	Human Resource	18	30

Jobs

job_id	job_title	min_salary	max_salary
10	Dep Manager	$120,000	$150,000
22	Accountant	$60,000	$80,000
33	Programmer	$60,000	$80,000
45	Senior Programmer	$70,000	$120,000

Locations

location_id	street_address	postal_code	city	state_province	country_id
10	123 Collins St	3000	Melbourne	VIC	1
20	222 Bourke St	3000	Melbourne	VIC	1
30	555 Swanston St	3000	Melbourne	VIC	1

Countries

country_id	country_name
1	Australia
2	Vietnam
3	Spain

Job History

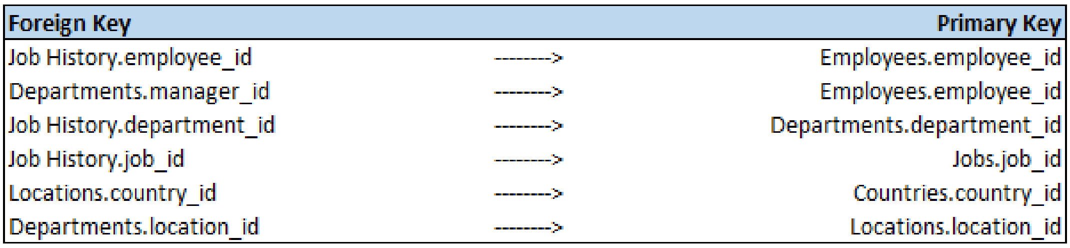
employee_id	start_date	end_date	job_id	department_id
10	1/01/2001	10/04/2002	33	1
10	11/04/2002	20/08/2002	33	1
12	1/01/1998	5/10/2003	33	1
12	6/10/2003	6/10/2004	33	1
12	7/10/2004	7/08/2009	33	1

Figure 2: Employee Database Instance

The database schema is shown below, and the meaning of most attributes is self-explanatory. "Job History" is simplified as "JobHistory". Primary keys are underlined, and foreign keys are annotated with a *.

```
Employees(employee_id, first_name, last_name, phone_number, hire_date, empjob_id,
salary, department_id)
Departments(department_id, department_name, manager_id*, location_id*)
Jobs(job_id, job_title, min_salary, max_salary)
Locations(location_id, street_address, postal_code, city, state_province,
country_id*)
Countries(country_id, country_name)
JobHistory(employee_id*, start_date, end_date, job_id*, department_id*)
```

The following table further clarifies the connection between the keys across multiple tables.



Foreign Key		Primary Key
Job History.employee_id	-------->	Employees.employee_id
Departments.manager_id	-------->	Employees.employee_id
Job History.department_id	-------->	Departments.department_id
Job History.job_id	-------->	Jobs.job_id
Locations.country_id	-------->	Countries.country_id
Departments.location_id	-------->	Locations.location_id

Figure 3: Keys

The following questions must be answered based on the given database schema and instance. Where explanation is required, each answer should be a SHORT passage of at most several lines.

QUESTION 4.1: Does the database schema ensure that there is a job associated with each employee? Explain your answer.

Questions 4.2: Can an employee work for two departments at the same time? Can an employee take two jobs at the same time? Explain your answer.

QUESTION 4.3: The Human Resource department has recently changed to have three sub-departments (i.e., Ongoing Staff Department, Casuals Department, and External Contractors Department). Now, each sub-department is supposed to have a separate manager. Temporarily and until the new managers are hired, Joseph has been assigned to the management of all three sub-departments.

The following SQL statements are intended to record all the changes required in the database instance. Will they work? Explain your answer.

```
INSERT INTO Departments VALUES(3, ‘Human Resource’, 18, 30);
INSERT INTO Departments VALUES(3, ‘Human Resource’, 18, 30);
```

QUESTION 4.4: The employee named Jonny Deans has recently been promoted to Senior Programmer. The following SQL statement intends to make the required changes in the database instance to reflect Adam’s promotion.

```
UPDATE Departments SET empjob_id=45 WHERE employee_id=10;
```

After running the above query, consider the request “find all the past contracts that Jonny Deans used to have”. Can this request be completed using the given database schema and after the above statement is run? If yes, explain how the request can be answered. If no, explain what is missing and how it should be fixed.

QUESTION 4.5: Explain what the result of executing the following SQL statement on the database instance will be.

```
 UPDATE LOCATIONS SET country_id=1,2 WHERE location_id=20;
```

QUESTION 4.6: Write an SQL statement to create the Jobs table including all the constraints. Make reasonable assumptions for the data type associated with each field. Your SQL statement must be valid for SQLite Studio environment and free of any errors.

QUESTION 4.7: Write an SQL statement to create the Employees table including all the constraints, assuming all the tables that Employees depends on already exist in the database. Make reasonable assumptions for the data type associated with each field. Your SQL statement must be valid for SQLite Studio environment and free of any errors.

QUESTION 4.8: A new employee has been hired. You are asked to update the given database instance so that it also includes the new employee. The new employee's ID is 100, his name is "Aladdin Brown", his contract will start at 1 July 2021, and he works in the Human Resources department. Other information about Aladdin's appointment will be finalised later. Your SQL statement must be valid for SQLite Studio environment, free of any errors, and compatible with your answer to the previous question.

Referencing guidelines

Use RMIT 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 format

You should submit one PDF document with all answers together.

You must use Lucidchart to work on Part 1 of your assignment. You may use Word or any other word processor to compile your submission. Use section titles to indicate which question you are answering. At the end, convert it into PDF format. Do not submit Word files. If that option is not available on your system there are free pdf converters online you can utilise (e.g. http://convertonlinefree.com/).

Submit the PDF file by the due date to the Canvas.

Academic integrity and plagiarism

Academic integrity is about honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge, and ideas.

You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e., directly copied), summarised, paraphrased, discussed, or mentioned in your assessment through the appropriate referencing methods.
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites.

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own.

RMIT University treats plagiarism as a very serious offence constituting misconduct.

Plagiarism covers a variety of inappropriate behaviours, including:

- Failure to properly document a source

- Copyright material from the internet or databases
- Collusion between students

For further information on our policies and procedures, please refer to the University website.

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.

Assessment declaration

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