THE UNIVERSITY OF SYDNEY

School of Information Technologies

COMP9120 Database Management Systems

Assignment 2: Logical Database Design

Group assignment (10%)

Introduction

This assignment is about the logical database design for an ER diagram included in this assignment description. The objectives are to gain practical experience in relational database schema creation, including integrity constraints, based upon a given entity-relationship (ER) diagram.

This is a group assignment for teams of about 3 members, and it is assumed that you will continue in your Assignment 1 group. You should inform your tutor as soon as possible if you wish to change groups.

Please also keep an eye on your email for any further announcements made on eLearning about this assignment.

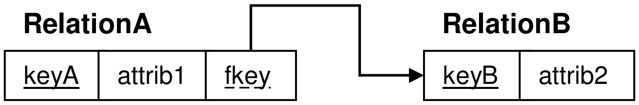
Submission Details

The final submission of your relational database schema is due at 5pm on the 23rd of September. You should submit the *items for submission* (detailed below) in a single zip file via eLearning.

Items for submission

Please submit your solution in the 'Assignment' section of the unit's e-learning site by the deadline, including the following items in a single zip file:

- Your submitted **SQL database schema** should be a text file (`.txt` or `.sql` file suffix) containing all DDL statements necessary to fully 举例说明 instantiate a working database, and DML statements to 增添数据 populate each relation. Your file should run without errors in Oracle 12c. You can annotate your statements using '`--`' at the start of lines of comment. You should group your statements for ease of reading (e.g., by keeping all table constraints within the relevant CREATE TABLE statement rather than declaring them externally, if possible).
- You can optionally include a **relational model (RM) diagram** to provide a visual model of your database schema. For this assignment, if included, the figure below summarises the syntax to use for the RM diagram, which should be submitted in PDF or jpeg format



- All team members must also confirm that none of their work is plagiarised by signing and submitting a cover sheet to their tutor for the final submission.

Marking

This assignment is worth 10% of your final grade for the unit of study.

Your group's submission will be marked according to the attached rubric.

Group member participation

If members of your group do not contribute sufficiently you should alert your tutor as soon as possible. The tutor has the discretion to scale the group's mark for each member as follows:

Level of contribution	Proportion of final grade received
No participation.	0%
Full understanding of the submitted work.	50%
Minor contributor to the group's submission.	75%
Major contributor to the group's submission.	100%

IMPORTANT: Policy relating to Academic Dishonesty and Plagiarism.

All teams must declare that the work is original and not plagiarised from the work of others. In assessing a piece of submitted work, the School of IT may reproduce it entirely, may provide a copy to another member of faculty, and/or communicate a copy of this assignment to a plagiarism checking service or in-house computer program. A copy of the assignment may be maintained by the service or the School of IT for the purpose of future plagiarism checking.

Design Brief: Relational Database Schema for a Restaurant Booking System

Your task is to create a relational database schema for the entity-relationship diagram that is shown on page 4. In particular your solution should include:

- Tables and attributes with suitable data types to capture all information in the model;
- Appropriate key constraints (PRIMARY KEY, UNIQUE, FOREIGN KEY) for all tables;
- Correct foreign key specifications including ON DELETE clauses where suitable;
- Appropriate additional integrity constraints expressed by means of NOT NULL, DEFAULT or CHECK clauses;
- ? INSERT statements to populate each relation with at least one record, to demonstrate a database instance consistent with the ER model.

Additional details

In addition to the model shown in the ER diagram, the following details apply: Not null

1. Fields in a tuple related to dates and times should always have values.
2. End dates should not occur before start dates (eg: booking end date before booking start date should not be allowed).
3. Bookings should be for at least one person (NumberOfPeople attribute on Booking).

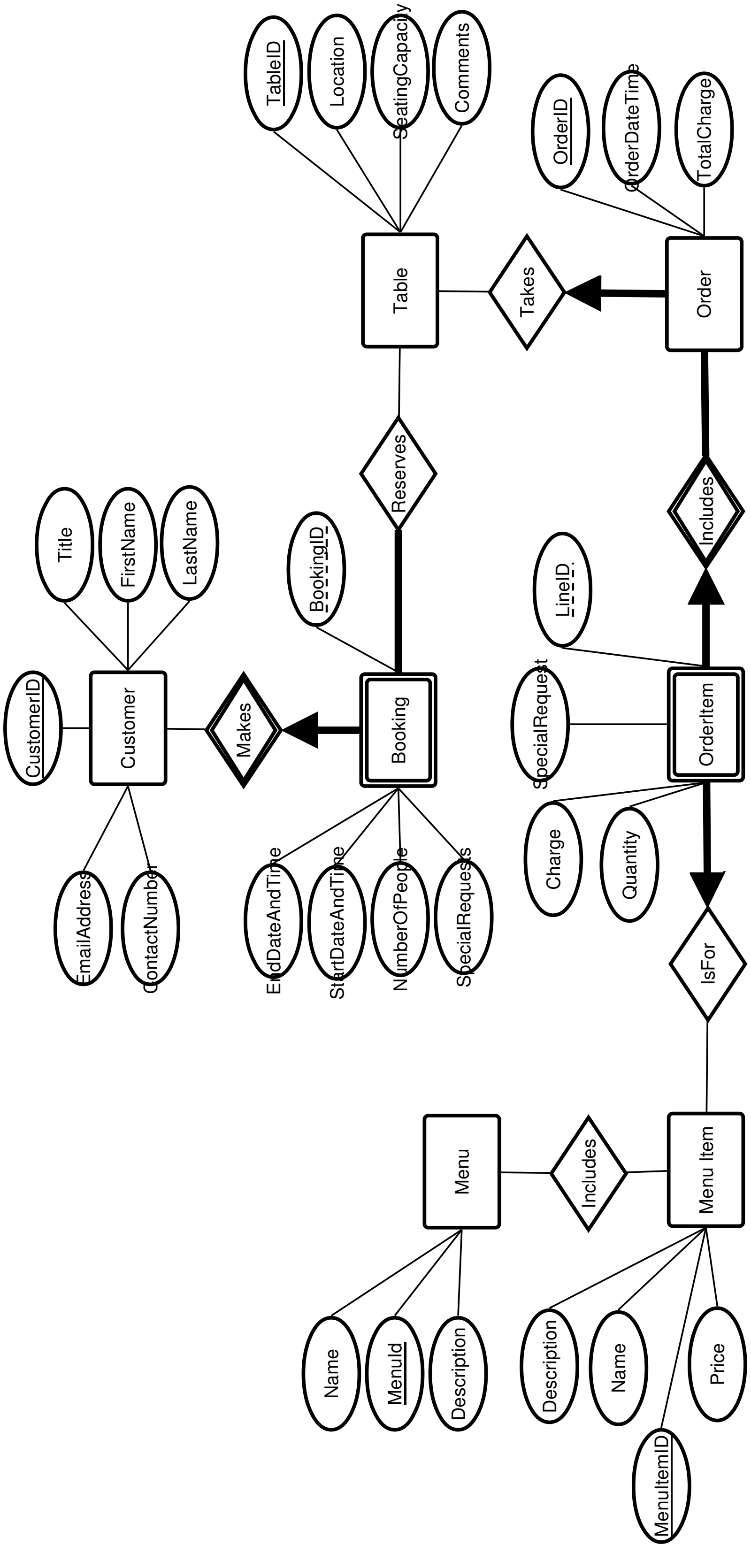
CHECK NumberOfPeople > 1

4. All fields in a tuple relating to details about a name (eg: Menu Item Name, First Name, etc) should always have a value. Not Null

5. Customers must have a specified contact number. Not Null

6. The total charge for an order, the charge for an order item, and the price of a menu item should always have values. Not Null

ER Diagram Shown On Next Page


Customer
CustomerID
Title
FirstName
LastName
EmailAddress
ContactNumber
Makes
Booking
BookingID
EndDateAndTime
StartDateAndTime
NumberOfPeople
SpecialRequests
Reserves
Table
TableID
Location
SeatingCapacity
Comments
Takes
Order
OrderID
OrderDateTime
TotalCharge
Includes
OrderItem
LineID
SpecialRequest
Charge
Quantity
IsFor
Menu Item
MenuItemID
Description
Name
Price
Includes
Menu
MenuId
Name
Description

Extension:

Please ensure that you have covered the non-extension requirements before attempting the extension. Marks are not considered for the extension unless full marks for non-extension requirements are achieved.

For Distinction/High Distinction level submissions, proficiency can be demonstrated through appropriate use of the following features:

- a non-trivial trigger definition supporting a constraint that cannot be enforced through the methods already listed above; See below for one sample trigger option.
- advanced domain constraints, e.g., using regular expressions;

Non-Trivial Trigger Option

If you decide to implement a non-trivial trigger for Distinction / High Distinction level submissions, you may implement a trigger based on the requirements below, or a trigger of equal difficulty:

One sample option for a trigger: The trigger should only allow insertions or updates for a given booking, such that each inserted or updated booking is associated with tables that have a seating capacity to fit the number of people specified on a booking. The booking insert or update should otherwise be rejected.

- Your model will be considered correct if you can show that your constraints (including trigger(s)) can prohibit a transaction from successfully committing when:
    - You create or update a booking tuple with PeopleCount greater than the sum of SeatingCapacity from tables associated to the booking.
- To simplify the specification, you do not need to consider how updates to fields on relations other than Booking affect this constraint (eg: you do not need to consider if the constraint is violated due to updates made to Table SeatingCapacity).

Marking Rubric

Your submissions will be marked according to the following rubric, with a maximum possible score of 10 points.

	Novice (0 pts)	**Competent (1 pt)**	**Proficient (2 pts)**
Relational Mapping	less than competent schema of the given scenario	all main entities and relationship mapped correctly to relations, with reasonable choice of data type for most attributes	the core model was very well mapped to a relational schema and good choice of data types for all attributes
Key Constraints	no key constraints captured at all	some primary and foreign keys were defined, but either incorrectly or incomplete	all necessary primary keys and foreign keys given including some useful ON DELETE and/or On UPDATE clauses
Semantic Constraints	no further integrity constraints given	some integrity constraints such as CHECK, DEFAULT or NOT NULL were defined, but either incorrectly or incomplete	all necessary integrity constraints for the model were given
Example Data	No example data given or yielded multiple errors	Some table examples missing or generated an error.	Database fully populated with a consistent set of data.
Advanced features	No advanced features supplied	One small demonstration provided, or larger example with some problems.	At least one very good example supplied.

Escaping Oracle Keywords in DDL

If you need to escape oracle keywords like “Table”, you will need to use double quotes.

Eg: CREATE TABLE "Table" (...);