Mock Coursework Assignment

Question, Model Answer, and Marking Scheme

Instructions

The coursework below should be answered by using your account on Oracle. Please log into Oracle and make sure your account is working. The coursework is divided in two parts, namely Part I and Part II. You need to answer Part I first. You have 50 minutes to answer this part and submit it on the submission area of Moodle. After this, you will receive the questions for Part II and instructions on how to run a script that will provide you with the environment to answer Part II questions. You will have 20 minutes to answer and submit the answers for Part II in Moodle.

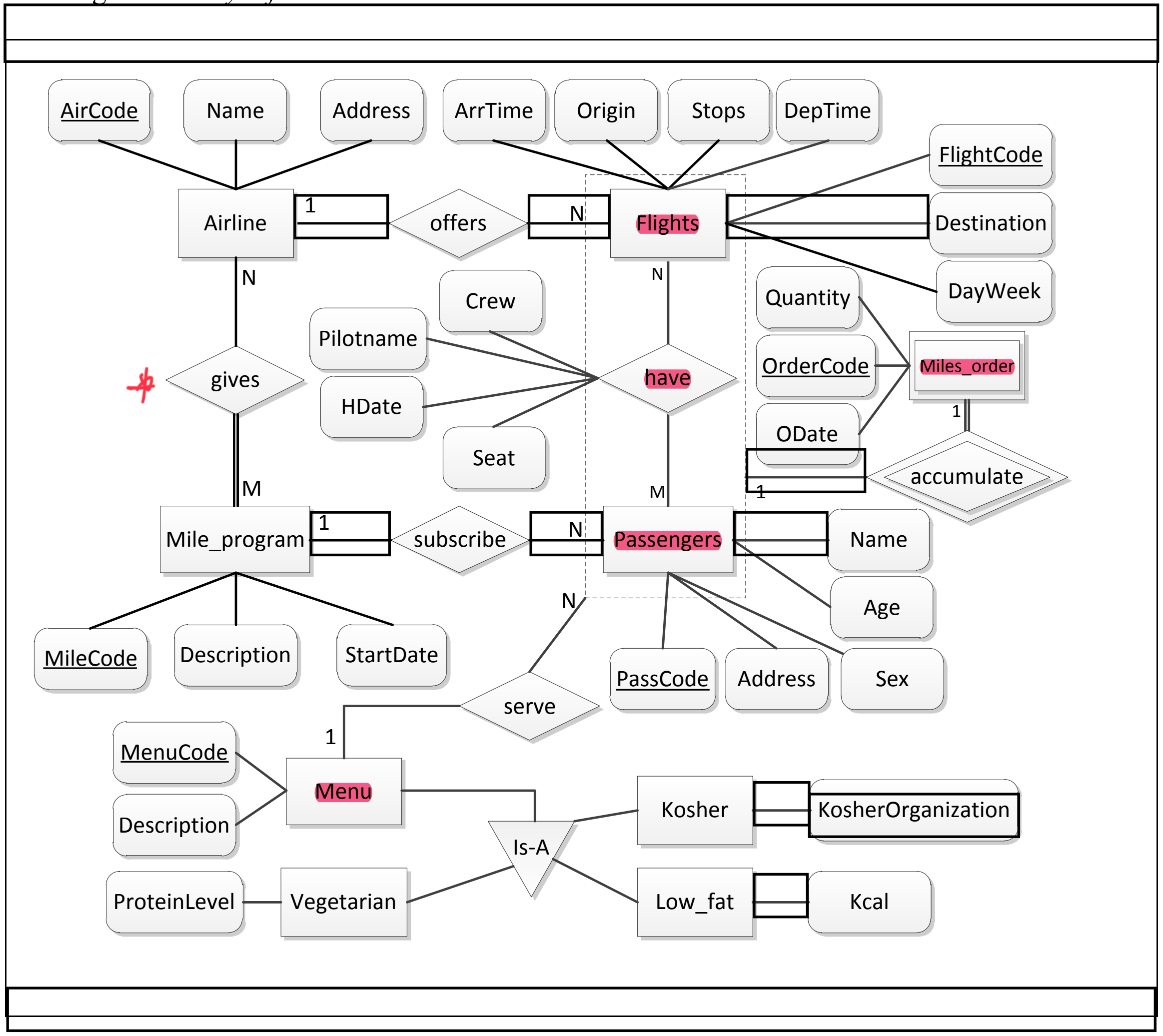
For both Part I and Part II, we suggest you write the answers using a text editor and use this file to submit your answers on Moodle. You can use the Oracle database to test your answers before submitting them.

Mock Coursework Assignment

Part I

Consider the ER diagram below for a database system for an online travel agent named *FlySafe.* The requirements of the database are also presented below for completeness and to facilitate understanding of the ER diagram. Based on the ER diagram, create an object relational database system (ORDBS) using Oracle SQL3.

ER Diagram for *FlySafe* database



Notes:

(i) All date values are in the format of DDMMMYY (e.g., 25DEC11)

(ii) All time values are in the format of HH:MM (e.g., 22:10)

Description of requirements for *FlySafe* database

The system is composed of many airlines. Each airline is identified by a code, name and head quarter's address. An airline has different mile programs, identified by a code, description, and starting date. The different types of flights offered by an airline have a unique number, day of the week, departure time, arrival time, origin, destination, and stops (if applicable). Passengers are allocated into flights and are identified by a number, name, address, sex and age. A passenger can subscribe for one mile program of a certain airline. The system records information about the passengers booked in a flight, together with the date, seat, pilot-name, and crewmembers of that flight. The system also maintains information about the miles accumulated by each passenger in a certain flight. This information is kept as a mile-order identified by a number, quantity and date. For each passenger in a flight a menu is served. The different menu options have a code and a respective description. The menu can be of type vegetarian, low fat, and kosher. For each vegetarian and low fat menu, information about the protein level and Kcal, respectively, of the meal is recorded. For the kosher type menu, information about the kosher regulation organisation is recorded.

Mock Coursework Assignment

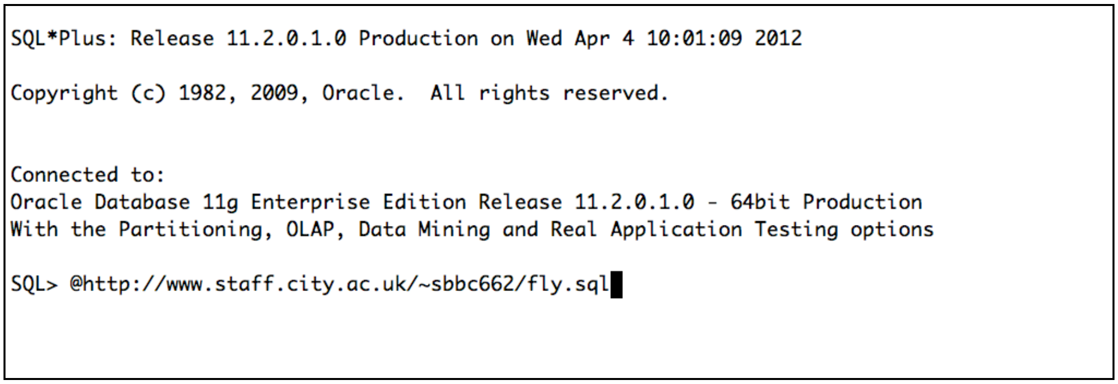
Part II

Instructions to set up the *FlySafe* database in SQL*Plus.

After logging in to SQL*Plus type the following command at the SQL> prompt:

@http://www.staff.city.ac.uk/~sbbc662/fly.sql

It should look like this:



```
SQL*Plus: Release 11.2.0.1.0 Production on Wed Apr 4 10:01:09 2012

Copyright (c) 1982, 2009, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> @http://www.staff.city.ac.uk/~sbbc662/fly.sql
```

Press Enter on the keyboard and the script will create the types and tables described by the schema.

Notes:

This script will drop any tables or types you have created with the same names as those shown in the model answer you have been given (see below).

You will receive warning messages that some types and tables do not exist in the run of the script. This is expected behavior.

This script will not affect any tables you have created with names that do not match those in the model answer. When writing your queries **YOU MUST USE THE TABLE AND COLUMN NAMING AS SHOWN IN THE MODEL ANSWER**.

At this moment you should have the database schema for the FlySafe database described in Part I instantiated with values in your Oracle area. A copy of the database schema is presented below for your reference. Please answer the following questions in Oracle SQL3. You should submit your answers on Moodle in your submission area. You have 20 minutes to answer these questions and submit the answers on Moodle.

(a) List the airline name that passenger John Wood has taken on the 04/APR/2012. (20 %)

(b) List the description of the mile programme of passengers that live in London (20 %)

SQL3 Scheme Model Answer: ปา type , เขียน Table

เหลือง Attribute

```
CREATE TYPE AirlineType AS OBJECT (
   AirCode      VARCHAR2(5),
   Address    VARCHAR2(30),
   Name        VARCHAR2(15));
/

CREATE TABLE Airline of AirlineType (
  AirCode PRIMARY KEY);

CREATE TYPE MileProgrammeType AS OBJECT (
   MileCode    VARCHAR2(5),
   Description     VARCHAR2(30),
   StartDate       VARCHAR2(8));
/

CREATE TABLE MileProgramme of MileProgrammeType (
  MileCode PRIMARY KEY);

CREATE TYPE GivesType AS OBJECT (
   MileProgGives     REF MileProgrammeType,
   AirlineGives        REF AirlineType);
/
```

มีเพราะ N:M

```
CREATE TABLE Gives of GivesType (
*  Scope for (MileProgGives) is MileProgramme,
   Scope for (AirlineGives)     is Airline);
```

เอาไว้เชื่อม M,N

```
CREATE TYPE FlightType AS OBJECT (
   FlightCode VARCHAR2(8),
   DayWeek    VARCHAR2(10),
   DepTime    VARCHAR2(5),
   ArrTime       VARCHAR2(5),
   Origin          VARCHAR2(15),
   Destination   VARCHAR2(15),
   Stops            VARCHAR2(45),
   Offers      REF   AirlineType);
/
```

```
CREATE TABLE Flights of FlightType (
  Scope for (Offers) is Airline,
  FlightCode PRIMARY KEY);
```

1-N

```
CREATE TYPE PassengerType AS OBJECT (
  PassCode   VARCHAR2(5),
  Name        VARCHAR2(30),
  Address     VARCHAR2(45),
  Age         NUMBER(3),
  Sex         VARCHAR2(1),
  Subscribe    REF MileProgrammeType);
/
```

ฝั่ง N Ref ไปหา 1

"REF MileProgrammeType" means that the "Subscribe" attribute is a reference (REF) to another user-defined data type named "MileProgrammeType". This implies that the "Subscribe" attribute is a foreign key that refers to the primary key of the "MileProgrammeType" table.

ตัวเชื่อมไปหา MileProgramme

```
CREATE TABLE Passenger of PassengerType (
   Scope for (Subscribe) is MileProgramme,
   PassCode PRIMARY KEY);
```

ตัวเชื่อมไปหา MileProgramme

```
CREATE TYPE MenuType AS OBJECT(
   MenuCode        VARCHAR2(5),
   Description  VARCHAR2(20)) NOT FINAL;
/

CREATE TABLE Menu of MenuType (
 MenuCode PRIMARY KEY);

CREATE TYPE VegetarianType UNDER MenuType (
  ProteinLevel   VARCHAR2(20));
/

CREATE TABLE Vegetarian of VegetarianType;

CREATE TYPE LowFatType UNDER MenuType (
  KCal VARCHAR2(20));
/

CREATE TABLE LowFat of LowFatType;

CREATE TYPE KosherType UNDER MenuType (
  KosherOrganisation VARCHAR2(20));
/

CREATE TABLE Kosher of KosherType;


CREATE TYPE MilesOrderType AS OBJECT (
   OrderCode    VARCHAR2(10),
   Quantity  NUMBER(5),
   ODate       VARCHAR2(8)) ;
/
```

```
CREATE TABLE MilesOrder of MilesOrderType (
  OrderCode    PRIMARY KEY);
```

N:M

```
CREATE TYPE HaveType AS OBJECT (
  FlightHave        REF FlightType,
  PassengerHave  REF PassengerType,
  MilesOrderHave  REF MilesOrderType,
  MenuHave       REF MenuType,
  HDate  VARCHAR2(8),
  Crew  VARCHAR2(50),
  PilotName VARCHAR2(30),
  Seat  VARCHAR2(5));
 /
```

1,1 וכו׳ N→1

```
CREATE TABLE Have of HaveType (
  Scope for (FlightHave) is Flights,
  Scope for (PassengerHave) is Passenger,
  Scope for (MilesOrderHave) is MilesOrder,
  Scope for (MenuHave) is Menu);
```

Mock Coursework Assignment

MODEL ANSWER

Part I – The scheme above   (60%)

Part II

(a) List the airline name that passenger John Wood has taken on the 04/04/2012. (20 %)

```
Select h.FlightHave.Offers.Name
From Have h
Where h.HDate=’04APR12’ and
        h.PassengerHave.Name=’John Wood’;
```

(b) List the description of the mile programme of passengers that live in London (20 %)

```
Select p.Subscribe.Description
From Passenger p
Where Address like %London%;
```

Mock Coursework Assignment

MARKING SCHEME

Part I – (60%)

Types – 2.0 % for each type => 11 types * 2.0 = 22.0 %

Tables – 1% for each type => 11 tables * 1.0 = 11.0 %

Use of UNDER – 1.0% for each UNDER => 3 UNDER * 1.0 = 3.0 %

References – 1% for each reference => 8 references * 1 = 8%

Scope – 1% for each scope => 8 scopes * 1 = 8%

Primary Key – 1.0% for each primary key => 6 Primary Keys * 1.0 = 6.0 %

Use of NOT FINAL – 2.0 % => 1 Not Final * 2.0 = 2.0 %

TOTAL = 22 + 11 + 3 + 8 + 8 + 6 + 2 = 60 %

Part II – (40%)

a) 20 % => 10% for Select and From parts and 10% for Condition part
b) 20 % => 10% for Select and From parts and 10% for Condition part