|  |  |
| --- | --- |
| Assessment Title | Assignment (Southern Airport Maintenance Service) |

## Competency Details

|  |  |
| --- | --- |
| Unit code/s and title/s | ICTDBS506 – Design Databases |
| Qualification code/s and title/s | ICT50720 - Diploma of Software Development |
| Business unit/Work group | BARTS/IT Studies |

## Instructions

|  |  |
| --- | --- |
| Method/s of assessment | Product (Written and Observation) |
| Overview of assessment | The assessment has five parts. Each part has multiple tasks. The assessment includes determining the database requirements, developing the database through the conceptual, logical and physical design processes. It also involves the design of database security, database backup and recovery. Students need to document the database design according to organisation document template. The last part is to implement the database in application. |
| Task/s to be assessed | The assessments include the following:   * Determine the database requirements (Oral Communication & Report) * Conceptual design * Logical design * Physical design, security, backup & recovery. * Implementation of database in application |
| Time allowed | Refer to your schedule for submission dates |
| Location of assessment | Assessment can be completed anywhere with access to the resources required and where the conditions replicate noise levels and interruptions that people typically experience working in the ICT industry. |
| Decision making rules | To receive a satisfactory outcome for this assessment you must complete all parts correctly. |
| Assessment conditions | This assessment must be undertaken where the conditions replicate noise levels and interruptions that people typically experience working in the ICT industry.    This is unsupervised assessment, and you may access any required resources. |
| Resources required | To complete this assignment, you will need to use MySQL Server, MySQL Workbench, Oracle Database Server hosted in AWS and Oracle SQL Developer, NetBeans and word processor. Windows based machines are provided in your practical classes. You can use a Mac if you prefer but these are not provided in the classrooms. |
| Result notification and reassessment information | You will be provided feedback and the result for your assignment on TAFESA Learn. You will be and given the chance to resubmit with required corrections only once.  Refer to the TAFE SA assessment policy for more information <https://www.tafesa.edu.au/apply-enrol/before-starting/student-policies/assessment> |

**Assignment (Southern Airport Maintenance Service)**

**Scenario:**

Southern Airport Maintenance Service (later named as SAMS) officials decided that they need a new database to manage the information of aircrafts, technicians and the aircraft maintenance services. The maintenance unit carries out airworthy tests for aircrafts which landed on the airport. The existing database does not meet the requirements of the aircraft maintenance services. You are working with IT Works as the database administrator. IT Works got the contract to help SAMS to develop the new database.

The SAMS database administrator (Joe Brown) confirmed the scope of the database project will store the aircraft information which are stationed and tested at the Southern airport. It also stores the human resources information such as the technician information, trainings and the aircraft test details. This database project will not implement the database tasks such as data conversion, operation nor intensive performance tuning. They will be regarded as out of the scope at this stage.

The development of the new database is split into multiple parts:

* Determine database requirements from user needs
* Develop conceptual model
* Develop logical model
* Develop physical design, security design and backup & recovery
* Implement the database with a prototype

Initially, the management aims at using MySQL as the database. This project requires you to go through the database development life cycle such as collecting user-needs, deciding the functional requirements of the database. Define the database conceptual model, logical model and physical model. Create the database schema to show the table structures required. Estimate the size of the database. Design the user security and the backup recovery procedures. At end implement the database with a prototype for signing off the database design.

**Background information:**

The initial analysis already conducted 3 months ago has captured the following information:

Southern airport offers qualified technicians for carrying out airworthiness tests to aircrafts which landed in the airport. Therefore, the airport maintenance service unit needs the database to store information such as:

* The technicians and their qualified trainings
* The airline information
* The aircraft manufacturers
* The aircraft models and their technical information
* The aircraft specific information
* The aircraft test details

Each country has many airlines. Each country is identified by a country id and has country name. Each airline has a certain home country where it is flying from.

Each airline is identified by an airline ID (2 characters) which is the IATA-Code. It is a unique code. The airline has an airline name. The system needs to keep track of the airline information with their address (such as street, suburb, postcode, state and country), contact phone and website address.

Each airline may own many aircrafts. Each aircraft must be owned by one and only one airline.

An aircraft manufacturer is identified by a manufacturer id, and has information of manufacture name. An aircraft may manufacture many aircraft models. Each model must be made by one manufacturer.

Each aircraft model has a unique model id (e.g. ’A380’ or ’B737’) and is produced by a specific manufacturer (e.g., ’Airbus’, ’Boeing’ or ‘Gulfstream’). The aircraft model id normally contains some indication of the manufacturer. e.g. model name starts with a ’B’ is used by the manufacturer called Boeing. Model name starts with a ‘A’ is used by the manufacturer called Airbus.

A sub-model is a variant of an aircraft model. Each aircraft model must have one or many sub-models. Each sub-model must belong to one unique model (e.g. ’A380-800’ or ’B747-100’). For example, a ’B737-900ER or B737 MAX - 10’ is the extended range version of the latest version of the Boeing 737. Another example, a ’A380-900’ a specific variant of an Airbus A380. It has 650 seats in a standard configuration compared to 555 seats on the A380-800. Aircraft submodel is identified by the aircraft submodel id and has information of submodel description.

Each aircraft is identified by an aircraft id which is a serial number (6 digits) given by Southern Airport inhouse. All aircraft fly in Australia must also have an aviation registration number (ARN) given by the Civil Aviation Safety Authority. These registration number is commonly known and consists of a two-letter code identifying the owning airline and a three-letter unique id within the airline fleet. Both parts are separated by a hyphen. For example, ’VH-EAF’ would be a valid aircraft registration number. ’VH’ is the code for the Qantas fleet of aircrafts and ’EAF’ identifies one particular aircraft of the Qantas fleet. Most airlines give their aircraft a name. For example, as mentioned ’VH-EAF’ is named ’City of Adelaide’ by Qantas.

Every aircraft must be built according to one aircraft sub-model specification. Every aircraft sub-model has many aircrafts.

Due to runway safety issues, the airport maintenance unit needs to keep track of the technical details of each aircraft sub-model such as length, height, wingspan area, max payload weight, max cruising speed & maximum range. Note: the technical details for B737-800 are different with B737-MAX in the length.

**Example of a sub-model:**

|  |  |
| --- | --- |
| **Sub-model:** | B737-800 |
| **Engine Models:** | CFM56-7B |
| **Length:** | 39.5m |
| **height:** | 12.5m |
| **Wingspan Area:** | 35.8m |
| **Maximum Payload Weight:** | 20,540 kg |
| **Maximum Cruising Speed:** | 838 km/h |
| **Maximum Range:** | 5, 436 km |

|  |  |
| --- | --- |
| **Sub-model:** | B737 MAX -10 |
| **Engine Models:** | CFM-LEAP-1B |
| **Length:** | 43.8 m |
| **height:** | 12.3 m |
| **Wingspan Area:** | 35.9 m |
| **Maximum Payload Weight:** | 18,778 kg |
| **Maximum Cruising Speed:** | 839 km/h |
| **Maximum Range:** | 6,110 km |

**Example of an engine model:**

|  |  |
| --- | --- |
| **Engine Model Name:** | CFM56-7B |
| **Made by:** | CFM International |
| **Thrust range:** | 121 kn |
| **Dry Weight:** | 2,370 kg |

Each aircraft sub-model is designed to use one engine model. Each engine model may be used by many aircraft sub-models.

Therefore, each engine model is identified by the engine model name and has information such as made by thrust range and dry weight.

Each aircraft may be mounted with zero, one or more engines. Each engine must belong to an engine model. Each engine model may build 0, 1 or many engines.

Each engine may or may not be mounted on an aircraft while the aircraft is at service. Each engine that mounts on an aircraft is identified by an engine no.

An aircraft may be mounted with zero, one or many engines. For example an aircraft of B737-800 has mounted with two engines. Each engine has an identification number such as: CFM56-7B20-340-131-712-0-17846. Some manufacturers may have 30 characters engine identification number. Each engine must have a manufacturing date.

Each aircraft sub-model must be operated under an aircraft category e.g. either as a passenger category or a cargo category. Most technical details are in common for both of those categories. A passenger category or cargo category is a kind of sub-model. A passenger category has a maximum number of passengers.

For example: Airbus B737-800 passenger category with two classes is configured to have 144 seats in economy class and 16 seats in 1st class.

A cargo category (e.g. the ’B737-800 ’) is designed to carry only cargo and has a maximum cargo weight 23,587 kg. Main deck offers 141.5m3 and the lower deck has 43.7m3 of space.

**B737-800 Passenger plane:**

|  |  |
| --- | --- |
| No of 1st class passengers: | 16 |
| No of economy class passengers: | 144 |

**B737-800 Cargo plane:**

|  |  |
| --- | --- |
| Main deck capacity | 141 M3 |
| Lower deck capacity: | 43.7 M3 |
| Maximum cargo weight: | 23,587 kg |



https://onemileatatime.com/singapore-airlines-737/

For technician information, each technician is uniquely identified by their employee id which is an automatically increased serial number. The system is also required to store the name (both first and last name), address (street, suburb, postcode), phone, salary and their login name. All technician’s salary must be either zero or a positive value. All technician first name and last name must have a value (i.e. they cannot be a null value).

Each technician is may have trained for zero, one or more aircraft models (but not more than three models). Each training is identified by a training id and has information of training name, and training date. Each training may train many technicians. The system is required to record which technician has completed which trainings (i.e. complete or incomplete).

Each aircraft model has 0, 1 or many qualified technicians.

This information about technicians must also be recorded.

Each technician may be supervised by 0, or one supervisor. A supervisor may supervise 0, one or many technicians.

The manager is a kind of technician. The manager has all the information of the technician plus the extra information such as the date that he/she becomes a manager. An ordinary technician does not have this extra information. Note: the job nature of a manager is not the same as the supervisor.

There are many kinds of tests that the Civil Aviation Safety Authority (CASA) to ensure that aircrafts are airworthy. Each kind of test is identified by a unique test id and has information of a test name. Each kind of test consists of 1 or many test items. Each test item is identified by an item code and has the information of the name of the test item. Each test item must belong to one kind of test.

Each aircraft may book for many test events. Each test event must be booked by one aircraft. Each test event is identified by the test event id, the date/time start the test, the date/time end the test, the number of hours the technician spent test, result of the test, and the result comments. The result of the test event must be either be null, “Pass” or “Fail” in value.

Each test event carries out only one specific kind of test. Each kind of test may be conducted by 0, 1 or many test events. The CASA requires the airport to keep track of each aircraft being tested under a certain test event supervised by a manager.

Each test item in a test event must be recorded with which technician is responsible for the test item and also test result (i.e. Pass or Fail). Note, a test event may need several technicians to carry out.

The manager has concerns in the availability of the database system particularly on the backup and recovery procedures. The new database must ensure that all data are backed up regularly. Data must be available 7 \* 24. Most database transactions are generated evenly everyday during the week even during the weekend. Two third of the transactions happen between 6:00am ~ 8:00pm. SAMS has already got the database backup infrastructures (such as the backup storage and the network) for the existing system that can also be used for the new database. The manager needs IT Works to determine the requirements of backup and recovery for the new database. The manager has indicated that if any data are corrupted, the database must be able to recover the data to the time before the corruption.

**Requirements**

Given the Southern Airport Maintenance Service (SAMS) database project, you are required to carry out the following tasks:

**Part 1 (Determine the database Requirements – Oral Communication & Report)**

Conduct a meeting with the client SAMS Database Administrator Joe Brown (your lecturer will impersonate the Joe Brown which is the user representative for the manager, human resource personnel and the technicians) to analyse the database functionality. You **must** participate in the requirement workshop which is carried out in Session 3. Before the workshop, you need to read the background information of the SAMS scenario and watch the requirements video which is available in the assessment section of LEARN. In the workshop, you will meet the client to confirm the user needs and the database requirements. During the meeting, you need to use the oral communication techniques such as listening and responding to the client request.

**During the meeting.**

1. Raised at least two questions to seek for clarification in the context of the user needs and database requirements.
2. Give at least two responses when the client elaborates/addresses the database requirements.

**After the meeting**,

Submit a word document to confirm & report the user needs and the database requirements:

1. List at least 2 user needs from the SAMS manager.
2. List at least 2 user needs from the technicians.
3. List at least five functional requirements of the database which is concluded from the needs of the user.
4. List the phases of the database life cycle involved in the SMS database project.
5. List at least two design documents that you will produce at each phase of these database life cycle involved.

**Part 1 submission:**

**During the meeting:**

Student must attend the requirement workshop to have the oral communication assessed. For student who enrolled in the external delivery mode, they must arrange with the lecturer for an online collaborate session or attend the IT\_Prac session online with the lecturer.

**After the meeting,** submit a word document (Task c to g) through **LEARN** to confirm & report the user needs and the database requirements. The document must include:

* the name of the client representative i.e. SAMS database administrator
* the requirements listed for Part c to g.
* the statement requesting for the client feedback.
* Your name and title at the signature

**Part 2 (Conceptual Design)**

Use MySQL Workbench to produce a **conceptual data model** (i.e. conceptual ERD) for the database. The conceptual ERD must incorporate the following requirements:

Note: The conceptual ERD must not include the non-key attributes

1. The database must be named as “**samsdb**”.
2. You mustconfigure the MYSQL Workbench to use the crow’s foot notation.
3. Identify the entities required to meet the functional requirements. Note: All entities must be named singular.
4. Assign a primary key to each entity identified.
5. Create relationship with the correct entities.
6. Assign cardinalities to each relationship.
7. Show the foreign key constraint with the name is generated by the workbench. The foreign key must be shown as the caption.
8. Name the relationship with a verb and must be shown as the second caption.
9. Use the naming convention specified which is specified in the document (Naming Convention of Database.docx)
10. Send an email to request the client to give feedback on your conceptual model whether those entities and relationships meet the user needs.

The email must include the subject tile. The email content must describe the purpose of the email as asking for confirmation of the user needs and functional requirements. The email must address to the right person (i.e. your lecturer) and the email must have your email signature.

1. Wait for the client feedback. Respond to the feedback and make changes before moving into the logical design.

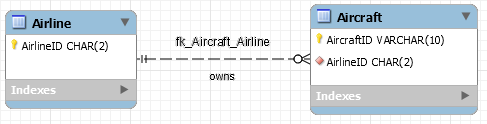
**Part 2 Submission:**

* Submit a screen shot of the whole ERD diagram in the word document.
* Submit the MySQL Workbench conceptual ERD diagram to **LEARN**.
* An email to the client SAMS database administrator (impersonated by your lecturer) asking for the feedback for your conceptual model. The email must include:
  + the name of the client representative i.e. SAMS database administrator
  + the statement requesting for the client feedback.
  + Your name and title at the email signature
* Make changes from the feedback and seek further feedback (no more than 3 rounds) before moving into the logical design.

**Part 3. (Logical Design)**

1. Each relationship in the ERD can be described with a pair of business rules.

For example, the relationship between the airline and the aircraft entities:



The pair of business rules for the above relationship can be described as the following:

* + Each airline owns 0,1 or many aircraft(s).
  + Each aircraft must be owned by one and only one airline.

Choose at least 8 relationships, write their business rules that shows both their cardinalities (i.e. maximum occurrences) and participation constraints (i.e. minimum occurrences).

Submit the business rules (for a least 8 relationships) in a Word document to LEARN.

1. Further develop a logical data model (i.e. logical ERD) of the database from your conceptual data model which has been signed off in Part 2 by the client.

In your logical ERD, pay attention to the following design issues.

1. Identify / refine the required entities. All entities must be named singular.
2. All entities must have a primary key (i.e. can be single column or multiple columns). Decide to use surrogated primary key or not. Decide the primary key to be auto increment or not?
3. Create / refine the relationships with the correct entities.
4. All relationship must have captions for both the foreign key constraint and the relationship name. All relationship names are capitalised the first latter and camel case. The relationship name must be a verb.
5. Determine /refine the cardinality & participation constraints according to the business rules for each relationship.
6. Use non-identifying relationship / identifying relationships appropriately.
7. Add attributes to the entities. All attributes must meet the need of the business operation.
8. Each entity must be normalised. i.e. all non-key attribute must be full functional depending on the primary key. Attributes must not have multi-values.
9. Foreign keys must have the same data type with the primary key of the referenced table.
10. Apply data validate rules to attributes:

* Assign data type to attributes. The type and size must meet the need of the business operation.
* Each non-key attribute must be considered whether null or not null for the integrity of the business operation.
* Each non-key attribute must be also considered with default, unique, sign/unsign values.
* Each non-key attribute must be also considered whether limited values applies or not i.e. Enum( ). Implement the test result can only have the value of “Pass” and “Fail”.

1. Implement super-type / sub-type relationship in the ERD. e.g. Manger is a kind of technician. A technician may not be a manager.
2. Implement recursive relationship in the ERD. e.g. Each manger may supervise zero one or many technicians. Each technician may be supervised by 0 or one manager.
3. Separate complex attributes so that they can be accessed independently. e.g. Name attribute becomes FirstName, LastName. Similarly for Address attribute need to break down into separate Street, Suburb, … City, Country.
4. Apply system constraints to the attributes. Consider the parental constraints and referential integrity appropriately (i.e. cascade update & delete issues). e.g. Delete the record of the parent entity also delete the related records of the child entity.The management would like the business operation when a test event is deleted, all of the test item in the test event also be deleted automatically

1. Generate and execute the create table database schema sql script. Submit the generated sql script and also provide a screen shot in the word document to show all tables are created as tables in the database.
2. Document the database structures.
3. Map the ERD to logical data structure. List **all** tables in the following relational logical data structures. Note: must indicate the primary key PK (in bold) and foreign key FK (in italic)

For example, the AircraftModelTechnician table should be presented as: AircraftModelTechnician (***AircraftModelID, TechnicianID***, QualifiedDate)

1. Document at least **three** tables using the following format of Relational Database Schema.

**Relational Database Schema:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Field Name** | **Data Type** | **Relational Description** |
| AircraftModel | aircraftModelID  manufacturerID | INT  INT | Each AircraftModel must be one or many aircraft.  Each AircraftModel must be one or many subModels  Each AircraftModel must have one and only one Manufacturer |
| EngineModel | engineModelName  manufacturer  thrustRange  dryWeight | Varchar  Varchar  INT  Decimal | Each EngineModel must have one or Engine |
| Passanger | PassangerCatagoryID  subModelID  maxPassangers  maxFirstClassPassangers  maxEconomyClassPassangers | VARCHAR  VARCHAR  INT  INT  INT | Passanger must have one and only one SubModel |

1. Document at least **three** attributes using the following format of Attribute Description.

|  |  |
| --- | --- |
| **DATA DICTIONARY: – ATTRIBUTE DESCRIPTION** | |
| SYSTEM : SAMS | DATE: 13/05/2023 |
| AUTHOR : Andre Alexandrov | PAGE: 1/1 |
| ATTRIBUTE NAME: maxPassangers | |
| ALIAS (Synonym) : N/A | |
| DATA SOURCE: Passanger | |
| **DATA STRUCTURE** | |
| Type: INT | |
| Length and Format: INT max value of 4294967295 | |
| **Characteristics** | |
| Range of Values  There is no specified range so goes by INT max of 4294967295 | |
| DESCRIPTION  The INT value describes the maximum number of passengers a passenger model plane can carry irrespective of class | |
|  | |
| **DATA DICTIONARY: – ATTRIBUTE DESCRIPTION** | |
| SYSTEM : SAMS | DATE: 13/05/2023 |
| AUTHOR : Andre Alexandrov | PAGE: 1/1 |
| ATTRIBUTE NAME: dryWeight | |
| ALIAS (Synonym) : N/A | |
| DATA SOURCE: EngineModel | |
| **DATA STRUCTURE** | |
| Type: DECIMAL | |
| Length and Format: max length of 6 with 2 digits reserved for after the decimal point | |
| **Characteristics** | |
| Range of Values  0000.00 – 9999.99 | |
| DESCRIPTION  A decimal describing the dry weight of an engine model | |
|  | |
| **DATA DICTIONARY: – ATTRIBUTE DESCRIPTION** | |
| SYSTEM : SAMS | DATE: 13/05/2023 |
| AUTHOR : Andre Alexandrov | PAGE: 1/1 |
| ATTRIBUTE NAME: manufacturer | |
| ALIAS (Synonym) : N/A | |
| DATA SOURCE: EngineModel | |
| **DATA STRUCTURE** | |
| Type: VARCHAR | |
| Length and Format: string with a max length of 45 | |
| **Characteristics** | |
| Range of Values  Any string shorter than 45 characters in length | |
| DESCRIPTION  Value to define who the manufacturer of the specific engine model is. | |

1. After receiving the conceptual design ERD, management has found that there are increasing number of aircrafts under their maintenance have been changed ownership to different airlines at different dates. The manager has changed the business to:

“Each aircraft must be owned by 1 or many airlines. Each airline may own 0, 1 or many aircrafts.” The business has a need to record the history of the aircraft being owned (i.e. dateOwned) by which airline. Incorporate this change to your design.

Submit a screen shot of the modified logical ERD.

1. Send an email to request the client to give feedback on your logical data model whether those entities, relationships, attributes meet the user needs. The email must include:
   * the name of the client representative i.e. SAMS database administrator
   * the statement requesting for the client feedback.
   * your name and title at the email signature
2. Wait for the client feedback. Respond to the feedback and make changes before moving into the physical design.

Submission requirements for Part 3:

* Tasks of Part 3 (a to e) with a zipped file including the following:
* A word document with answers of the tasks includes:
* Pair of business rules for at least 8 relationships. &
* Documents of the data structures (relational logical data structures for all tables, relational data schema – 3 tables, & 3 attribute descriptions)
* Screen shot of the logical ERD diagram.
* Screen shot showing the list of tables created after the successful execution of the sql script.
* MySQL Workbench model file with the logical ERD.
* The generated database schema sql script file.
* An email to the client SAMS database administrator (impersonated by your lecturer) asking for the feedback for your logical design model.
* Make changes from the feedback and seek further feedback (no more than 3 rounds) before moving into the physical design.

**Part 4. (Physical Design, Security and Recovery)**

1. One of the aspects of the physical database design involves creating indexes. Indexing affects database performance. Indexes are best used on the columns that are frequently used in the foreign key attribute, the WHERE clause & attributes need sorting with ‘order by’ in the sql queries. The primary keys are automatically indexed. However the foreign keys are not automatically indexed.

You are required to create indexes on the foreign keys for the following 3 tables.

* AircraftModel
* AircraftSubModel
* Aircraft

Note: you need to use the naming convention when creating indexes.

For submission provide the CREATE INDEXES statement and the screen shots of the execution of these statements.

1. The management has additional physical requirements that the following query will be used frequently and hence appropriate indexes to good performance:

*“Display all the aircraft ids and the technician names for those technicians have carried out the name of the test item as* ***Engine start and run up according to AFM procedures*** *with the test event during* ***January 2022*** *for those aircrafts being tested”*.

Display the output with the ascending order of test date/time, aircraft ID, last name and first name.

1. Write SQL statement for the above query.
2. Compare your existing logical data model, give a list of attributes of which table(s) should be indexed in order to ensure the performance of the query for this additional requirements.

Hints:

* Some indexes may be a single attribute or sometimes may be multiple attributes.
* Consider indexes can be on the foreign keys, the WHERE conditions and the sorting orders.
* Note: No need to create a separate index if the foreign key is part of the primary key of the table since the primary keys are automatically indexed.

1. Write the CREATE INDEX statements to create the indexes as required in part (ii).
2. Execute the above CREATE INDEX statements and capture the screen shots to show those indexes have been added successfully in the database.
3. The client has a new requirement asking you to design an interactive data entry screen for user (technician) to enter aircraft test information of a test event which has been booked. Each aircraft must test regularly for air worthiness. You are required to design an input screen for the user (technician) to enter the aircraft test results during the test event. (Note: you may use any prototyping tools to design the screen layout).

This screen can be broken down into the following queries to support this requirement.

1. After the screen had displayed with the TestEventID, the user will enter the AircraftID (e.g. aircraftid.Text), then the Airline ID, the aircraft name, aircraft model and sub-model will be displayed on screen for reference.
2. The user will enter the information of the test start date and the manager id, the system will display the manager name.
3. The user will enter the kind of test (i.e. IATA test id), the system will populate the test item codes and their description on screen.
4. The user will enter the test item results. Each test item can either be ‘pass’ or’ fail’. The corresponding technician no. will be entered against the test item that they have conducted and the date of the signature signed. The user will also enter the overall test result (Pass or Fail). When the submit button is clicked, the following tables will be inserted:

* TestItemTestEvent record will be inserted
* TechnicianTestItemTestEvent record will be inserted
* TestEvent record will be updated

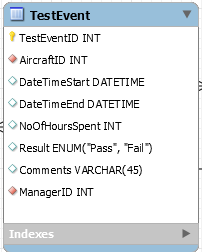
Submit your design of the data entry screen in the word document.

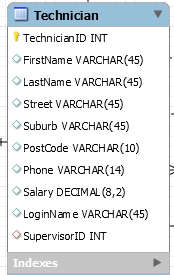
1. You are required to design the queries needed to support the screen for data entry of the test result in Task C.
2. For comparing whether the database has sufficient data to support the query in Task C (i), indicate this by writing the SQL statement with attributes come from the tables required.
3. For comparing whether the database has sufficient data to support the query in Task C (ii), indicate this by writing the SQL statement with attributes come from the tables required.
4. For comparing whether the database has sufficient data to support the query in Task C (iii), indicate this by writing the SQL statement with attributes come from the tables required.
5. For comparing whether the database has sufficient data to support the query in Task C (iv), indicate this by writing the SQL INSERT statement with attributes come from the tables required.

In your query design process, you may experience that your conceptual data model may have tables or attributes that are not considered in the previous stage of the design process. Indexes are not created to support better performance for the above queries. In the following table, evaluate/compare whether your design up to this stage meets the user needs or not.

|  |  |
| --- | --- |
| Need to add/drop tables, attributes, indexes: | Reason |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

1. You are required to estimate the size of the following tables with indexes.





In order to avoid the repetition of calculation for the size of the whole database, you are required to estimate the size of the above two tables. Assume there are 100 Technician records and 1000 TestEvent records in the database. Estimate the size of these tables and their indexes. Show the steps of your calculation.

Note: In your calculation, for simplicity, no need to factor for the housekeeping for BTREE or Tablespace Fragmentation.

TestEvent

((4 + 4 + 8 + 8 + 4 + 1 + 4 + 46 + 4) + (6 + 4) + (6 + 4)) \* 1000 = 103000

Technician

((4 + 46 + 46 + 46 + 46 + 11 + 15 + 9 + 46 + 4) + (6 + 4))\* 100 = 28300

1. The database requires multiple-user access. You are required to implement the following database user access and security features (i.e. user permissions). The following show the role of the users of the database.

Human Resources (Jake Green):

* Maintain fully on the Technician and Training records.
* Maintain fully on the records of the technicians who have completed which trainings of aircraft models. i.e. TechnicianAircraftModelTraining table

Technician (Jim White)

* Able to display, insert and update records on the AircraftModel table. i.e. no delete permission.
* Display records qualified trainings completed.

Note: Technicians role does not maintain the Technician data at all.

1. Connect as the root user, use SQL Control Language statements to create the following users.

* User is **‘Jake’** with the password **‘Green’**
* User is ‘**Jim**’ with the password ‘**White**’

1. Use SQL Control Language statements to create the following two user roles.

* HumanResources
* Technicians

1. Use the root user to insert at least 2 records (with meaningful data) to **all** tables in the database. Note: You need to pay attention to the referential integrity issues during the insertion of the records.

Submit all INSERT statements of your sample data in a script called **sams\_insert.sql**; remember to provide the screen shots for the sql output of the insert statements.

1. In the following table, determine what user permissions (i.e. use **C,R,U,D** as their access rights allowed) are needed for the user roles.

Where C for insert

R for read

U for update

D for delete

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Technician table | AircraftModel table | Training table | TechnicianAircraftModelTraining table |
| Managers | R | R | R | CRUD |
| Human Resources | CRUD | R | CRUD | CRUD |
| Technicians | R | CRU | R | R |

1. Use the root user to insert at least 2 records (with meaningful data) to **all** tables in the database. Note: You need to pay attention to the referential integrity issues during the insertion of the records.

Submit all INSERT statements in a script called **sams\_insert.sql;**

1. Write SQL statements to grant user permissions according to your access controls to the human resources and technicians' roles.

Submit screen shots to show each of the SQL statements and the output results.

1. Use SQL Control Language statements, grant the roles to the users and set their default role as described in Tasks (i) & (ii). Submit the SQL statements and their output results.
2. Connect the database as **Jake** (human resources),write SQL statements for the following database actions. Execute the statements. Are they executed successfully or not? Submit the SQL statements and their output results.
3. Insert a record (with meaningful data) into the **Technician** table.
4. Update the Technician record that you have just inserted with the Salary = 200,000.00.

Submit screen shots to show each of the SQL commands used and their output results.

1. Connect the database as **Jim** (technician)
   * 1. Display all records from the **TechnicianAircraftModelTraining** table.
     2. Update the record from the **Technician** table (with meaningful data) (i.e. you may update the record you have just inserted).

Submit screen shots to show each of the SQL statements used and their output results.

1. Given with the existing Southern Airport maintenance security plan (Southern Airport Maintenance Service Security Plan.docx), after the review, management has concerns with the security of user passwords.
2. Revise the security plan by adding the following three password security features:
3. Password expiration – require passwords to be changed every 90 days to the existing users.
4. Password reuse restrictions – prevent old password from being chosen again within a year (i.e. 365 days).
5. Password need verification – require when password change also need to specify the current password to be replaced.

Add the above password security features to the existing **Southern Airport Maintenance Service Security Plan.docx** document at the appropriate section.

1. Write sql statements to alter the existing user so that the passwords for **Jake Green** and **Jim White** to apply the three password security features described in Task 1) a), b) & c) above.

Submission requirements:

* Submit the revised **Southern Airport Maintenance Service Security Plan.docx**
* Submit the sql commands
* Submit screen shots for the result of each of the above sql commands being used. (Note: you may have referential integrity issues.)

1. The management has indicated that the ‘*technician qualified with which aircraft model*’ and the ‘*test event results*’ are sensitive information. They would like to have those attributes be encrypted.
2. Write INSERT statements to insert two records with the encryption on the two sensitive data using the AES encryption technique.

Hints: for simplicity, create two new tables to implement the encryption and decryption in a new database.

1. Use SELECT statements to show the data are encrypted. i.e. show the data are scrambled.
2. Use SELECT statements to decrypt the sensitive data. i.e. show the data are readable.

For submission on each task above, provide the sql statements and the screen shots of data being encrypted and decrypted.

1. Backup and recovery procedures

The management has understood that the backup of the database allows the data to be recovered when the primary storage failure due to medium failure, accidental deletion or malicious attack etc. You are required to suggest SAMS the follow ways to protect data from being lost:

1. List two ways of SAMS can back up the database physically.
2. List at least two ways of SAMS can back up the database logically.
3. There are some database tables that are relatively static in nature such as the aircraft model technical information and technician information which are quite important to the airport operation. The management would like to have the database backup regularly and recover quickly even they may not be up-to-date. The management indicated that the operation is tolerable if those information are three days old.
4. Write backup commands (i.e. in batch file) to conduct the full backup of the database including the following requirements:

* The backup file name must include the disk directory and with date & time as part of it.
* Suggest when to do this backup. (i.e. day/time)
* How often this backup method be repeated?

Capture the screen shots with the commands executed successfully.

1. Write a sql command to recover the technician table from the backup file. Hints: need to turn off the database referential integrity and drop the Technician table before the recovery of the Technician table.

Capture a screen shot the Technician table as been dropped.

Capture another screen shot with the sql recovery commands executed successfully.

Submission for Task h:

A separate word document integrated the backup and recovery procedures specified in the Task h (i), (ii), (iii), 1) & 2). The document must include the following:”

* The heading/ title of the document.
* Backup procedure:
* The backup commands used
* The screen shots of the backup command executed.
* The day/time and the frequency of backup.
* The backup file location.
* The recovery procedure:
* The screen shots of the backup command executed.
* The illustration of the table being dropped (i.e. table no longer exist)
* The illustration of the table being recovered.

1. Send an email to request the client Joe Brown (impersonated by your lecturer) to give feedback on your physical model and security design whether they meet the functional requirements. The email must include:
   * the name of the client representative i.e. SAMS database administrator
   * the statement requesting for the client feedback.
   * your name and title at the email signature
2. Wait for the client feedback (from your lecturer). Respond to the feedback and make changes if needed.

**Part 4 Submission:**

* Submit a word document with answers for the task a to task h.
* Provide the sql statements in a script file. The script file must have comments
* For answers with sql statements or commands, screen shots must be provided with the output of the execution.
* Submit the Backup and Recovery document for the SAMS database system.
* An email to the client SAMS database administrator (impersonated by your lecturer) asking for the feedback for your physical model.
* Make the required changes from the feedback.

**Part 5. (Implement the database in application)**

The Southern Airport management has heard about **Oracle database** has features database objects. The management wants to implement a prototype of a simplified version of the database using the Oracle OORDBMS extension techniques for testing purposes.

For this part, you will have an account in the **Oracle database** in Amazon Web Services (AWS) for the Assignment. Your user name should be your firstname attached with your First character of the last Name plus **Ass2**. e.g. For John Smith his user name will be:

Student: John Smith

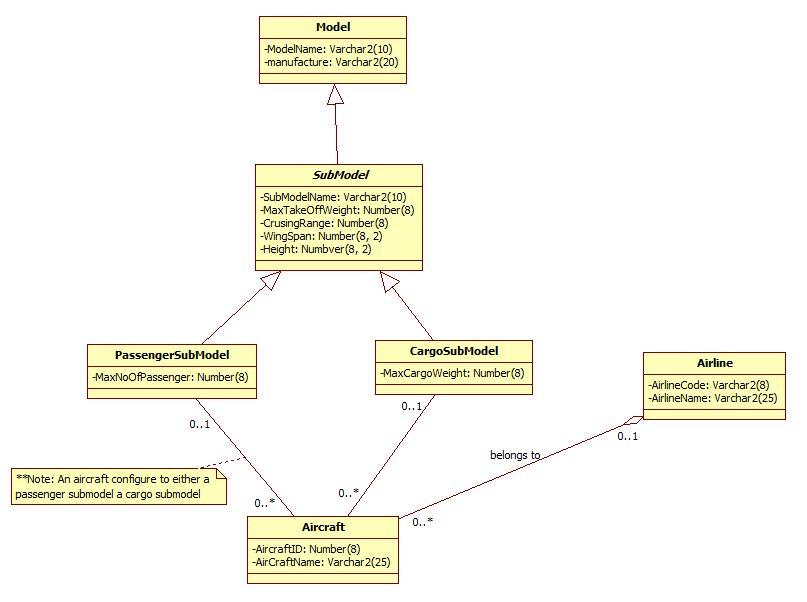
Oracle user name :  **johnSAss2**

The password : John Smith’s Student ID

The Oracle database server has a System ID (SID) called xe and it is located in the AWS VM server with a public IP address of 35.166.130.72. It uses the server port no. 1521.

Management prefers to keep the information as generic as possible. He confirmed the following business rules:

* A submodel is a kind of model. The submodel is an abstract class.
* A passenger submodel is a kind of submodel.
* A cargo submodel is a kind of submodel.
* Each aircraft may either belong to a passenger submodel or a cargo submodel.
* Each airline may have 0, 1 or many aircraft.
* Each aircraft must belong to one and only one airline.



Note: SubModel is an abstract class where it does not instantiate to an object instance.

Given the Airport Maintenance Scenario with the object model as show above, you are required to carry out the following tasks:

1. Create object type and table of objects. (Set up object ref)

Develop the database schema for implementing the model in the Oracle OORDBMS. You mustcreate the table using user-defined object types.

i.e. Make use of the super-type / sub-type (“**is-a**”), **REFs** (i.e. “**has-a** / **whole-part**” association) features of Oracle.

For simplicity, you only require defining the following object types:

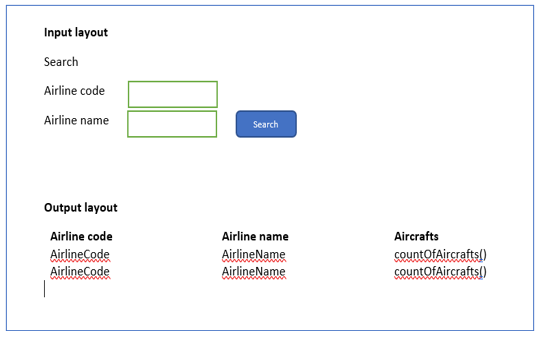
Airline, Model, Sub Model, PassengerSubModel, CargoSubModel, Aircraft

1. Create the above six Object Types. You are required to use the REF for association relationships. Create a new script file called **sams\_objects\_script.sql** for submission. Add your SQL statements to the script file.
2. Create Tables for their **object types.** Create Tables for their **object types.** Append your SQL statements to the script file.
3. Assign the primary keys and foreign keys for each object table. Append your SQL statements to the script file.
4. **Create database object methods and Insert object data**
5. Create a method in the appropriate object type to query how many aircrafts are owned by each airline. Append your SQL statements to the script file. Capture the screen shot that the member function has been created.
6. Insert three records (with meaningful data) into each table. Append your INSERT SQL statements to the script file. Run the database object method (i.e. the query) that you have designed with your test data. Capture the screen shots for successful execution.
7. **Run the database object method using the given example Java program (in NetBeans)**
8. Since the Oracle server is located in the AWS EC2 cloud server, provide the database network access details as your organizational access profile. Also provide the JDBC connection statement that you will use to connect to the Oracle database.

* Public DNS name:
* IP address / hostname:
* Port No:
* Region:
* Oracle SID database instance name:

The JDBC connection string:

1. Given the example Java program in NetBeans (Part5\_ASDS\_OracleConn.Zip) which has a form that has been designed to retrieve and display how many aircrafts are owned by each airline (i.e. Use JDBC connection) for validating your DB design. The form has the following layout.



Modify the database connection details in the program so that the program will use the designed database object method (completed I Task b(i)) to display the number of aircrafts owned by each airline.

1. **The client has the following question which need your clarification. Answer the questions in dot point. Summaries the key points in your own words:**
2. How can you identify redundant data in your design? List two ways.
3. With MySQL database, what kind of authentication can be applied? List at least 2 authentication methods.
4. In the logical design process, what design concepts did you employ when designing your data structures? In another words, why did you design your data structures that way? How the data structures influence the design of the screen or vice versa? (Approx in 50 words)
5. Given the concept of object modelling, how does the object modelling influence your design and how does it influence the screens and reports? (Approx in 70 words)
6. After the database has completed the performance tuning process (i.e. re-indexing), if the performance of the database is still not acceptable, describe two ways in scaling a database. (Approx 50 words)
7. **Final sign off.**

Send an email to obtain the final sign off from the client Joe Brown (impersonated by your lecturer). The email must include:

* + the name of the client representative i.e. SAMS database administrator
  + the statement requesting for the database design to be signed off.
  + Include the screen shot with the final prototype executing with the output.
  + your name and title at the email signature
* The client will give feedback to your submission, if there are any part that is not satisfactory, please respond to the feedback by resubmit your to LEARN again.

For the final sign off, you may assume that if the lecturer has marked your submission or resubmission as satisfactory in LEARN, the project is regarded as being signed off by the client.

**Part 5 submission**

* Submit a zip file with the word document (with screen prototype) for answers needed for all Tasks in a to d.
* Include the SQL scripts **sams\_objects\_script.sql** (which includes all create object type, create tables, create member function and insert records statements)
* Submit the whole NetBeans project & screen shots of the execution output for marking. The AWS VM server is for testing your codes. The Oracle server will be deleted at the end of its useable life span. Therefore, all submission must be uploaded to LEARN.
* Send an email to request the client Joe Brown (impersonated by your lecturer) to obtain the sign off the database design.