Module 1 Assignment

For each of the business use cases listed, choose one or more database architectures (relational, document, key-value store, columnar, graph, etc.) that appear to be a good fit.

1. A state Office of Emergency Services wishes to store a huge history of incidents within the state using structured data (dates, times, text strings and numeric values). There are nearly two hundred separate data items, but which ones are applicable varies considerably by the type of incident (e.g. wildfire, earthquake, tornado, flood, etc.)

Relational database can be used in this case. First of all, the data is structured so it can be fit into a relational database like SQL database. Second, a table can be created for each type of incident. And we can have different columns of each table to record the special characteristics of the incident.

1. A startup that hosts prescription drug information wishes to display medications and potential side effects, along with a visual presentation of how common the side effects are.

Graph databases can be used in this case. The data of side effects (including the name and percentage of patients having this side effect) can be stored in a graph format, with a link to the medication associated. Each medicine can connect to several side effects or medications. Due to these characteristics of the data, a graph database is suitable for storing this kind of data and presenting how common the side effects are.

1. A bank wishes to store and maintain bank account transactions with controls to prevent two business users from updating the same transaction at the same time, ad hoc query capability for resolution of customer service issues and audit requests, along with a rich set of security controls.

Relational database is suitable in this case. Since most of the relational databases are ACID(atomic, consistent, isolated, and durable), we can ensure that there are no two business users updating the same transaction at the same time. Also, using the feature of “view”(a virtual table based on a SQL statement) provided in many relational databases, ad hoc queries can be achieved by writing appropriate SQL statements.

1. A large retailer wants the ability to store and retrieve one or more image files for each product they sell, using the product ID as the identifier for the image files.

Key-value databases can be used in this case. Image files can be stored in Blob(binary large object), where the key-value DBMS does not need to know anything about the contents of the blob. The user just needs to store the blob using the unique identifier under a key-value DBMS, which is the product ID in this case.

1. A cellular provider needs to store massive amounts of circuit use information in the form of the beginning timestamp, duration, and assigned circuit. There are potentially millions of these circuit uses per minute. The data will be used as input to a process that will sift through the data, looking for trends in the data.

Columnar or document databases can be used in this case. Columnar database can split columns into column groups, allowing parallel processing of handling data and hence increasing the overall bandwidth. Document databases are also suitable for real-time analytics given its characteristics.

1. A provider of online data modeling tools wishes to store (and later retrieve) the metadata for each data model as XML and/or JSON documents.

Document or Key-value databases are suitable in this case. Since the value is stored in XML or JSON format, Key-value database is suitable for data stored in Blob. If we need to parse the content inside XML or JSON data in the data storing/retrieval stage, we can adopt a document database. It is capable of parsing document contents.

Module 2 Discussion:

* With all the GUI tools available for querying databases, why is SQL still important?
* What online applications have you used that likely have a large database? What are the problems with supporting large online databases?

SQL is still very important as it provides more functionality than GUI tools. For example, when a user tries to delete a row in the middle of a table with millions of rows in GUI tools, the user needs to use his mouse to scroll down to the specific location to complete the operation. With SQL, the user only needs to write a simple statement to match the row of data to be deleted. It is also capable of performing complex queries.

Some online applications that support billions of users should have a large database, like Facebook, Instagram, Google. The major problem of supporting a large online database is to enhance the scalability for supporting a rapidly growing number of users. Also, the ability to retrieve and compute from the large database is also important for the internet companies to recommend the advertisement to users.

Module 2 Assignment:

Music Inventory

Instructions

Using the Music Inventory database schema, code and execute queries for the following:

1. List the copyright year and title for each compact disc (CD), in descending order by copyright year.
2. List the CD ID and title for CDs copyrighted in 2014 that also have an in-stock inventory over 20.
3. List each copyright year along with the number of titles and the total quantity in-stock for the compact discs copyrighted that year.
4. List the CD ID and title of CDs that have an in-stock quantity that is higher than the average in-stock quantity for all CDs.
5. List the CD ID and title of all CDs where Stevie Wonder is an artist.

Downloads:

* [Setting Up a DBMS Schema](https://onlinelearning.berkeley.edu/courses/1667502/files/189079943/download?wrap=1)
* [Actions](https://onlinelearning.berkeley.edu/courses/1667502/assignments/22265510)
* [Music Inventory Database Documentation](https://onlinelearning.berkeley.edu/courses/1667502/files/188664723/download?wrap=1)
* [Actions](https://onlinelearning.berkeley.edu/courses/1667502/assignments/22265510)
* [Inventory Database (Oracle)](https://onlinelearning.berkeley.edu/courses/1667502/files/188664763/download?wrap=1)
* [Download Inventory Database (Oracle)](https://onlinelearning.berkeley.edu/courses/1667502/files/188664763/download?download_frd=1)
* [Inventory Database (SQL)](https://onlinelearning.berkeley.edu/courses/1667502/files/188664767/download?wrap=1)
* [Download Inventory Database (SQL)](https://onlinelearning.berkeley.edu/courses/1667502/files/188664767/download?download_frd=1)
* [Inventory Database (SQLite)](https://onlinelearning.berkeley.edu/courses/1667502/files/188664537/download?wrap=1)
* [Download Inventory Database (SQLite)](https://onlinelearning.berkeley.edu/courses/1667502/files/188664537/download?download_frd=1)
* [Inventory Database (Microsoft Access)](https://onlinelearning.berkeley.edu/courses/1667502/files/188664533/download?wrap=1)
* [Download Inventory Database (Microsoft Access)](https://onlinelearning.berkeley.edu/courses/1667502/files/188664533/download?download_frd=1)
* [Inventory Database (MySQL)](https://onlinelearning.berkeley.edu/courses/1667502/files/188664765/download?wrap=1)
* [Download Inventory Database (MySQL)](https://onlinelearning.berkeley.edu/courses/1667502/files/188664765/download?download_frd=1)

Please submit the SQL or Microsoft Access query along with the results returned when you ran the query.

Module 3

Normalization and Star Schema

* Application developers typically balk at normalization with arguments that the tables it generates make retrieving data too difficult and that the database will perform better if the need for joins is minimal. What counter-arguments can you offer?
* What are the main advantages and disadvantages of using a star schema design for a database that will store customer order history that will occasionally require updates to correct errors. (There are notorious data quality issues in the order system that supplies the data to the order history database.

Although it may be more difficult for SQL users to write a more complex SQL statement to retrieve normalized data, it can bring more benefits than harm for database users and administrators. For example, as introduced in lectures, there are three data anomalies of concern to database designers which are insert, delete and update anomalies. If any of the anomalies occurs, the amount of time to recreate and restructure the whole database is much longer than writing more joins during query stages. Therefore, although the normalization process may not be straightforward to database users, once users get used to it, they can realize that the normalized table is actually simpler to understand and easier to maintain.

Advantages:

Star schemas are straightforward to be understood by business users. Also, queries in star schemas are easier to write compared to 3NF schemas. Star schemas can be changed or expanded easily as the business requirement changes. For example, if the order method(e.g. online/ phone/ shop) needs to be added in the order history table, star schemas can be changed easily to cope with this change.

Disadvantages:

Since the data are occasionally updated, it requires more effort for the database maintainers to correct the error as data integrity is not enforced in a star schema design.

Module 4 Discussion

Discuss the main advantages and disadvantages of using a highly generalized design, such as the one presented Topic 4-1 using the Product Property intersection entity.

You are assigned to design the database for a major new application system. Discuss the advantages and potential disadvantages of immersing yourself in the process design of the application logic while being held responsible for the delivery of the database design in a timely manner.

Advantages of using a highly generalized design:

It is straightforward and simple for business users to understand and use. If database maintainers need to insert or update several rows, only one table needs to be involved. Therefore it is easy to write the correct SQL statement, therefore lower the operational risk.

Disadvantages of using a highly generalized design:

Using the product property example in topic 4-1, It is difficult to identify which attributes are applied to different products.  Therefore, most of the attributes need to be set as optional, allowing NULL values, and hence data integrity cannot be enforced. Also, more products may be added in the future, so the number of attributes will grow explosively.

Advantage of immersing in the process design:

The database designer can understand the business logic and requirements, so he/she can change the database design to address any potential problems.

Disadvantages of immersing in the process design:

More time is required for the database designer to walk through the application logic with other stakeholders, hence delaying the delivery of database design.

Module 5 Discussion

You are assigned as the database designer on a new project with an aggressive schedule. The application developers are in a hurry and they want to skip all the logical design steps and go straight to creating physical tables. What arguments can you offer to convince them to allow a little time for a logical design?

Understanding that we have an aggressive schedule, but the logical design of the database before the physical design is a critical step. If the logical design is done correctly with sufficient time, the design of the physical tables becomes very easy. It can save much time and resources for having the logical design compared to changing the physical implementation each time for different use cases. Skipping the logical design steps is more error prone as the logical design can reduce the possible error by removing anomalies and redundancies.

Module 6 Discussion

Security Challenges

Instructions

* What are the security risks if an application program co）nnects to the database using an account that owns the database schema?
* Some organizations implement database security by creating views that restrict the rows and columns that business users can see, and then granting business users access to the views instead of the underlying tables. Do you think this is adequate security? Why or why not?

It is very dangerous for an application program to own the database schema, as any user in the application may have a chance to alter, insert or delete the database, which poses a major security threat to the database. We should comply with the principle to give minimum but enough privilege to database accounts required by the application.

This is an adequate security method. As columns can be omitted to some users using view, so users cannot peek into important information in other columns, for example lookup code values. We can use a WHERE clause to limit returned rows, protecting data from being leaked. Of course, this method needs to work with a correct implementation of role and priviledge control. Always bear in mind that we should give the minimum but enough privilege to database users.

Module 7

When to use XML and object-oriented databases?

* Is XML becoming more or less important in the industry? What are the advantages and disadvantages of using XML for all data transfers, including those between two databases within the same organization?

In my opinion, XML is becoming less important in the frontend web industry as there is more technology and frameworks coming up, for example, REST API and Javascript/Typescript, so JSON or other formats may be more preferred in the industry. Nevertheless, XML still has a place in the backend application industry and this status shall remain for years. In many enterprise level backend applications, XML is still a standard for document transferral.

Advantages of XML:

XML is human readable based on its hierarchical structure (if indented properly).

It is highly transferable between different relational databases, and even different programming languages.

The tree structure of XML allows users to compare and combine the elements

Disadvantages of XML:

The efficiency of XML is worse than JSON and other data formats.

XML is also a verbose and redundant format, so it makes the file bulky and large.

It is more difficult to traverse an XML document compared to other formats.

* List some applications that you think could make good use of object-oriented or object-relational databases, along with the features you would implement using object technology. Please include your reasoning for the applications you chose.

Real-time applications can utilize object-oriented databases. I am working in finance, and we need to handle real-time trading data gathered from different exchanges. To ensure low latency and efficient queries for facilitating the trading strategies, object-oriented databases can be helpful in this case.

Moreover, some applications which require efficient calculation, and fast results are suitable to use object-relational databases. Examples include 3D modeling software like CAD, molecular science, and scientific calculation software.

Also, object-relational databases are capable of handling complex data, so it is applicable in biomedical fields, and large scale ecommerce applications.

Module 8

Business Intelligence Databases

What are the differences between how BI databases are used compared with databases that support everyday business transactions? Do you think normalization is more important in designing BI databases? Why or why not?

Business Intelligence databases are different with operational database. We can use the term defined by Ted Codd to describe these databases, which are online analytical processing(OLAP) and Online transaction processing(OLTP) databases to describe business intelligence databases and operational databases respectively. Some key differences between OLAP databases and OLTP databases can be summarized as follows:

* OLAP databases hold historical data while OLTP databases hold current data.
* OLAP databases store detailed data along with highly summarized data, while OLTP databases store each business transaction record separately.
* In OLAP databases, data is static, with new data added in a specific interval. While in OLTP databases, data is changing and new data is added whenever transaction is processed.
* As for database queries, the scope for accessing OLAP databases can be large and many rows of data need to be accessed. In OLTP databases, queries are normally short-running and only accessing few rows of data.
* As for target users, OLAP is designed for internal managerial users, while OLTP is designed for many concurrent users.
* As for transaction volume, OLAP databases are designed to support medium to low transaction volume, while OLTP databases need to support a much higher transaction volume.

Normalization is important in designing BI databases, but this statement is valid given that the source of data come from a normalized relational schema. If the prior condition is true, then the ETL(extract, transform, and load) specifications for migrating data to BI databases are easier to write and execute.

Also, it can reduce the ongoing maintenance cost as if new tables are added in the future, they can fit well with what has already been build, thanks to the advantage of normalization.

In my opinion, as a BI database contains historical data, so the data anomalies will rarely appear in a BI database. However, normalization does provide an additional advantage of providing a more flexible design. The normalized relations are easier to modify as business requirements change.

Module 7 Assignment

Recommend cursor processing options (SENSITIVE/INSENSITIVE, SCROLL/NO SCROLL, WITH HOLD/WITHOUT HOLD, FOR UPDATE) for the following database applications:

Online application for searching movie reviews

I recommend cursor to have following options:

SENSITIVE, SCROLL, WITH HOLD, READ ONLY

Reason for SENSITIVE option: Changes made outside the cursor should be immediately effective to the cursor. So that users can view any live update for the movie reviews when they are searching the database. In this way, users do not need to refresh their browser every time when a new comment is added. We can leverage this sensitive cursor to provide any update for the reviews to the user.

Reason for SCROLL option: SCROLL option allows the application to move through the review database in a non-sequential manner. When there are a thousand or million rows of review, we may need pagination on the UI for users to jump to different pages. In this case, SCROLL option is useful to display results at a different page.

Reason for WITH HOLD option: According to my user experience at several movie review websites, If a new review is added or an existing review is edited, it should not close the connection for other users when they are searching for reviews. They expect the website would not update unless they refresh on their side. Therefore, using a WITH HOLD option aligns with user expectation.

Reason for READ ONLY option: READ ONLY option specifies that rows retrieved using the cursor cannot be updated. In a searching application, almost all of the queries are supposed to only access and return data rows according to user’s requirement. So the cursor should not be updatable.

Online shopping application (searching products, adding them to an electronic shopping card, and then checking out).

I recommend the cursor to have the following options:

SENSITIVE, SCROLL, WITHOUT HOLD, FOR UPDATE

Reason for SENSITIVE option: With SENSITIVE option, changes made outside the cursor should be immediately effective to the cursor. An example to illustrate the importance of SENSITIVE option would be: When there is someone else who just bought the product and make the remaining inventory decrease to be 0, other users who are putting that product in their shopping cart should be able to know that this product is not available anymore. So SENSITIVE option is useful here.

Reason for SCROLL option: If there are a lot of products held in the shopping application, pagination or filtering functions are required for users to quickly accessing their desired product rather than requiring the user to view each product result row by row. So SCROLL option is useful here as it provides the ability to move through the database in a non-sequential manner.

Reason for WITHOUT HOLD option: Under WITHOUT HOLD option, after a transaction is committed, the cursor should be automatically closed. In an online shopping application, since there are some important functions which includes payment using credit card or other payment providers like Paypal. According to my understanding, payment should be an atomic operation, i.e. either completely success or fail. If we set the option to be WITH HOLD, then there may be some problems if we are accessing a failed transaction and incorrectly proceed with it.

Reason for FOR UPDATE option: Under FOR UPDATE OF <column list> option, rows retrieved using the cursor can be updated or deleted. In an online shopping application, there may be some functions need to update or delete some data rows, for example, after user complete the purchase of a product, the application needs to decrease the inventory number of the product. So FOR UPDATE can be used here.

Web-based application for maintenance of a company’s product database

SENSITIVE, SCROLL, WITH HOLD, FOR UPDATE

Reason for SENSITIVE option: With SENSITIVE option, changes made outside the cursor should be immediately effective to the cursor. In a web-based application for product database maintenance, if there are other transactions committed outside the cursor, the web UI should be able to reflect this change. Otherwise users may need to refresh the browser to initialize a new cursor for viewing this update.

Reason for SCROLL option: SCROLL option provides the ability to move through the database in a non-sequential manner. In the aforementioned application, as it needs to provide the function of pagination and filtering for users to view and update the product data efficiently, so SCROLL option is useful here.

Reason for WITH HOLD option: Under WITH HOLD option, after a database transaction is committed, the cursor should remain open. In this application, the possible database transaction would be update rows of product data according to different conditions. After committing the transaction, the cursor can remain open so that no need to redefine and reinitialize the cursor again.

Reason for FOR UPDATE option: The application involves updating or deleting rows of data to perform the maintenance of the product database, so FOR UPDATE option is needed here.

Application to maintain your personal music and playlist library

SENSITIVE, SCROLL, WITH HOLD, FOR UPDATE

Reason for using SENSITIVE option: If there are some transactions committed outside the cursor, it should be visible to the cursor. So that the application can show live update if any music/playlist is inserted/updated/modified.

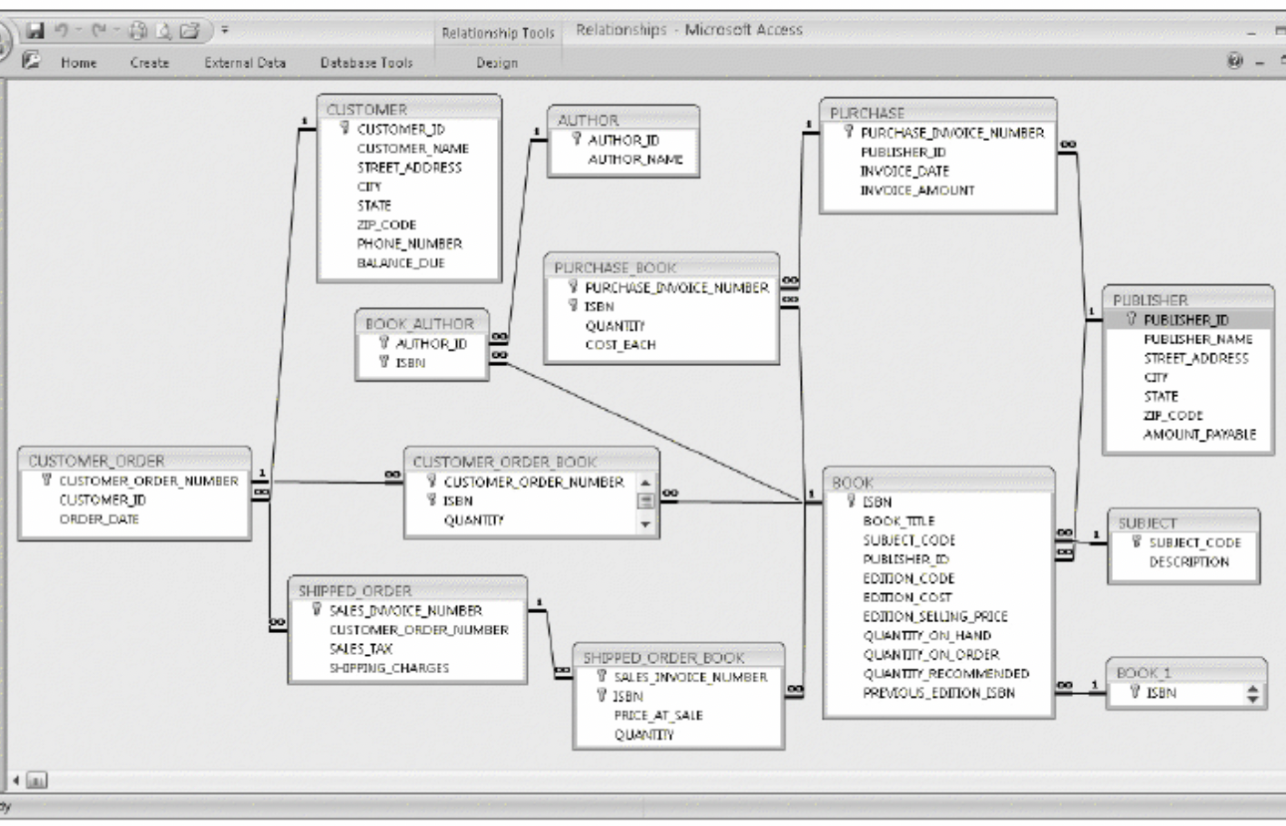
Reason for using SCROLL option: There may be some functional requirements from the users to access the data in a non-sequential manner. For example, user may want to play a random song in his/her playlist, or retrieve playlist in a different page. Therefore, using SCROLL option for the cursor is suitable.

Reason for using WITH HOLD option: WITH HOLD option allows the cursor to stay open after the transaction is committed. If a song or playlist is inserted/deleted, the cursor can remain open to perform the remaining transaction, so no need to reinitialize or redefine the cursor again.

Reason for using FOR UPDATE option: To enable the application to maintain the music and playlist library, the cursor needs to have edit rights on the data. Therefore, FOR UPDATE option is suitable in this example.

Module 3 Assignment

* For practice in normalization, complete *Try This* Exercises 6-1 and 6-2 from the textbook. (Solutions are in the textbook, so there is no need to turn in this part of the assignment.)
* Create a star schema model for the Computer Books Company database design shown in the solution to Try This 6-2 from the textbook (found in Appendix B beginning at the bottom of page 452). You should have fact tables for book inventory, publisher book purchases, and customer book orders.

­­­­­

Module 4 Assignment

Data Model

Instructions

Create a data model for a medium-sized business that uses the common patterns discussed in this module. Use an entity relationship diagram to document your model.  Your solution must be normalized to third normal form as described in Module 3 (Topic 3.2).  (Normalization is a best practice for databases used for business transactions.)

Include the following entities and attributes (listed in parentheses):

* Employees (Employee ID, Given Name, Middle Name, Family Name, Birth Date, Pay Grade Code, Pay Grade Name. Home Phone, Business Phone, Mobile Phone, Home Address)
* Customers:
* Individual Customers (Customer ID, Customer Given Name, Customer Family Name, Business Phone, Mobile Phone, Billing Address, Mailing Address, Shipping Address, Work E-Mail Address)
* Organizational Customers (Customer ID, Customer Name, Business Phone, Billing Address, Mailing Address, Shipping Address, E-Mail Address, Web URL)
* Suppliers, which can also be customers. (Supplier ID, Supplier Name, Business Phone, Mobile Phone, Billing Address, Mailing Address, Shipping Address)
* Organizational Structure
* Division (Division ID, Division Name, Division Manager Employee ID, Division Manager Name)
* Department (Department ID, Department Name, Division ID, Division Name, Department Manager Employee ID, Department Manager Name)
* Work Group (Work Group ID, Work Group Name, Department ID, Department Name, Work Group Leader Employee ID, Work Group Leader Name, List of Employees Assigned to Work Group)
* Contact Methods – Generalize all the phone numbers, e-mail addresses, and home/shipping/mailing/billing addresses into common Contact Methods.
* Purchase Orders (Supplier ID, Order Date, Purchase Order Number, Item Description, Quantity, Price Each, Item Subtotal [Quantity \* Price Each], Order Status Code)

Note:  All physical addresses include Address Line 1, Address Line 2, City, State/Province, Country, Postal Code

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address ID | Contact ID | Address Type Code | Address Line 1 | Address Line 2 | City | State/Province | Country | Postal Code |
| 10001 | 3669 | HOME | Rm 123 | 456 Happy St | Berkeley | CA | USA | 90090 |

[Data Modeling and Drawing Tools](https://onlinelearning.berkeley.edu/courses/1667502/pages/data-modeling-and-drawing-tools) available in both free and paid forms online

Module 5 Assignment

Physical Data Model

* Create a physical data model for the medium-sized business schema you created in Assignment 4. Include the following:
  + Data types for all attributes with NULL or NOT NULL constraint.
  + Resolution of any supertype/subtype relationships in the logical model (implement as-is, collapse into subtypes only, or rollup into supertype).
  + A list of any added indexes (beyond the primary key indexes) **must be included**.
* Which NoSQL database implementations do you think would be good choices for the schema described in the textbook in Try This Exercise 8-1 and why?  Please recommend at least two NoSQL database implementations.
* [Data Modeling and Drawing Tools](https://onlinelearning.berkeley.edu/courses/1667502/pages/data-modeling-and-drawing-tools) available in both free and paid forms online

Module 6 Assignment

Security and Precautions

* For the medium-sized business schema used in Assignments 4 and 5, write the SQL statements for the following:
  + Create roles for the Managers, HR Specialists, and Other Users.
  + Grant privileges to the roles as follows:
    - All users have read access to all the tables except Employment.
    - Managers have read access to all the tables (including Employment), plus the ability to apply inserts, updates, and deletes to all tables **except** Division, Department, Work\_Group, Pay\_Grade, Employment, and Employee\_Work\_Group\_Assignment
    - HR Specialists have read access to all tables (including Employment), plus the ability inserts, updates, and deletes to the Division, Department, Work\_Group, Pay\_Grade, Employment, and Employee\_Work\_Group\_Assignment   
      Hint: To simplify your task, assume you can grant roles to other roles

CREATE ROLE Manager; CREATE ROLE HR Specialist; CREATE ROLE Other User;

GRANT SELECT ON Customer

* List the security precautions you would recommend for each of the following database applications:
  + Company personnel database
  + Hospital medical records database
  + Bank checking account database
  + Consumer credit database
  + Automobile insurance claims database
  + Medical insurance claims database
  + Personal music library and playlist database

Module 8 Assignment

Designing a Data Warehouse and a Data Mart

* Design a data warehouse schema for the medium-sized business application used in Assignments 4, 5, and 6. A full history of changes for every entity (table) is required as described in the Topic 8.1 Lecture  Most data warehouses follow Summary Table architecture using history tables to track change history, but you may follow Star Schema architecture using Type 2 dimensions is you wish.
* Design a data mart (star) schema for your medium-sized business data warehouse schema. All dimensions must be Type 2, meaning a full history of changes is required as described in the Topic 8.1 Lecture under the heading "Handling History in Star Schema Data Warehouses.