Topic One

In this section you will cover the following topics:

Determine the requirements of developing queries

Recognise various query-related terminologies

Identify the type of data source for a chosen query language

Identify and use the necessary tools, and environment, in building queries

Database Management Systems

A database-management system (DBMS) is a collection of interrelated data and a set of programs to access those data. The collection of data, usually referred to as the database, contains information relevant to an enterprise. The primary goal of a DBMS is to provide a way to store and retrieve database information that is both *convenient* and *efficient*.

Database systems are designed to manage large bodies of information. Management of data involves both defining structures for storage of information and providing mechanisms for the manipulation of information. In addition, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access. If data are to be shared among several users, the system must avoid possible anomalous results.

Because information is so important in most organizations, computer scientists have developed a large body of concepts and techniques for managing data. These concepts and technique form the focus of this course.

Terminology

The following list of database terminology applies to the structure of data within a relational database and will be used throughout this document;

Database Definitions

Attribute	The characteristics of an entity.
Client	A single-user computer that interfaces with a multiple-user server.
Client/server database system	A database system that divides processing between client computers for data input and output, and a database server, used for data inquiries and manipulations.
Column	A field within a table.
Data modelling	The process of organizing and documenting the data that will be stored in a database.
Database	A collection of electronically stored organized files that relate to one another.
Database management system (DBMS)	A system used to create, manage, and secure relational databases.
Entity	Any group of events, persons, places, or things used to represent how data

	is stored.
ERD model	The Entity Relationship Diagram model is a representation of data in terms of entities, relationships, and attributes.
File	A collection of similar records.
Foreign key	A column in a table that links records of the table to the records of another table.
Keys	Columns of a table with record values that are used as a link from other tables.
Normalization	A three-step technique used to ensure that all tables are logically linked together and that all fields in a table directly relate to the primary key.
Primary key	A column in a table that uniquely identifies every record in a table.
Referential integrity	A system of rules used to ensure that relationships between records in related tables are valid.
Relational database	A collection of two or more tables that are related by key values.
Relationship	An association between entities.
Row	A record within a table.
Server	A multiple-user computer that provides shared database connection, interfacing, and processing services.
Table	A two-dimensional file that contains rows and columns.

Historical Perspective

Before the existence of the computer-based database, information was transcribed on paper and stored in a physical file. Ideally, each file contained a separate entity of information, and was most commonly stored in either a file cabinet or card catalogue system.

An organization that stored files in this manner may have, for example, had one file for personal employee information and another file for employee evaluations. If the organization needed to update an employee name, each individual file for the employee needed to be updated to maintain consistent data.

Updating files for one employee was not a big deal, but if several employee names needed to be updated, this process was very time consuming. This method of storage not only called for multiple updates among individual files, but it also took up a great deal of physical space.

With the advent of computers, the information in the files moved to databases, but the format for the databases continued to mirror the hard copy records. In other words, there was one record for each piece of information. The problems with associated hard copy records were also mirrored. Using the example above, if an employee's name needed to be updated, each individual file of the employee had to be updated. On the other hand, searching for information was considerably faster and storage was more centralized. Files of this type are referred to as "flat" files since every record contains all there is about the entity. (SQL for MS Access)

Relational Database

A modern database is a collection of organized files that are electronically stored on a server. The files in a database are referred to as tables. We come in contact with databases every day. For example all Banks use databases which people interact with via ATMs, and when we purchase products from large retail organisations there are computer-based systems which are updated. The most popular and widely implemented type of database is called a relational database. A relational database is a collection of two or more tables related by key values.

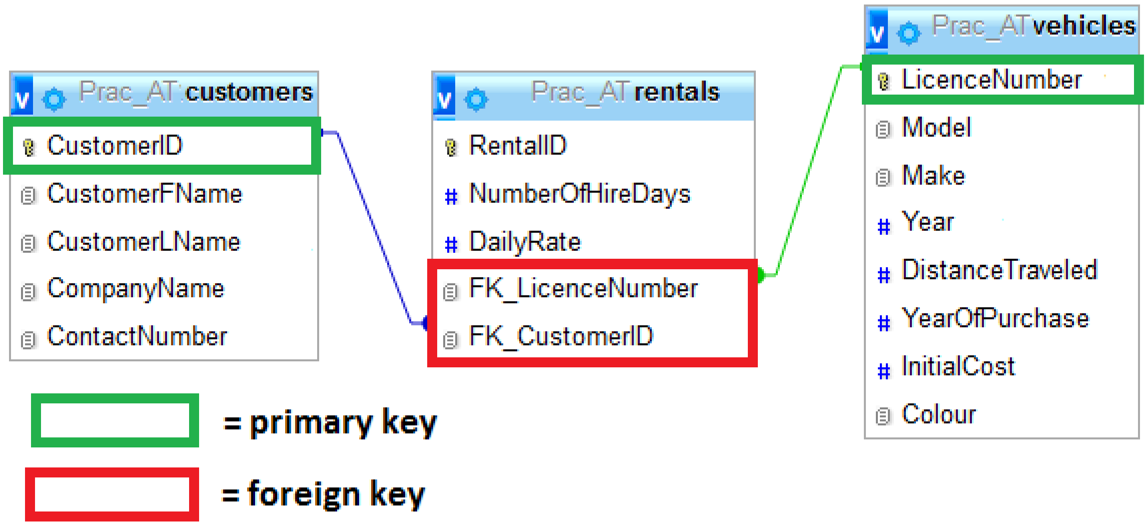
Tables

We refer to the tables in a database as two-dimensional files that contain rows (records) and columns (fields). Each row in the table represents an individual record and each column represents an individual entity of information. For example, a table named Customers could have the following seven entities (columns) of information: First Name, Last Name, Address, City, State, Zip, and Phone. Each customer entered into the Customers table represents an individual record.

Keys

To create a relationship between two tables in a relational database will use keys. Keys are columns in a table that are specifically used to point to records in another table. The two most commonly used keys during database creation are the primary key and the foreign key.

The primary key is a column in a table that uniquely identifies every record in that table. This means that no two cells within the primary key column can be duplicated. While tables usually contain a primary key column, this practice is not always implemented. The absence of a primary key in a table means that the data in that table is harder to access and subsequently results in slower operation. On the other hand, tables with very few entries will often not be indexed. This is especially true if the value is not used for searches or lookups.



The foreign key is a column in a table that links records of one type with those of another type. Foreign keys create relationships between tables and help to ensure referential integrity. Referential integrity ensures that every record in a database is correctly matched to any associated records. Foreign keys help promote referential integrity by ensuring that every foreign key within the database corresponds to a primary key.

IMPORTANT: the rentals table has a primary key “RentalID”, however; it is not part of a relationship.

Normalization

Another widely implemented technique used in the planning stage of database creation is called normalization. Normalization is a three-step technique used to ensure that all tables are logically linked together and that all fields in a table directly relate to the primary key.

In the first phase of normalization, you must identify repeating groups of information and create primary keys. For example, the following column names represent columns in a database named Car Rentals:

CustomerFName	CustomerLName	CompanyName	ContactNumber	NumberOfHireDays	DailyRate	LicenceNumber	Model	Make	Year	DistanceTravelled	YearOfPurchase	InitialCost	Colour

Notice that the column names contain repeating groups of information and there is no primary key assigned. To complete the first form of normalization, eliminate the Customer related fields since they represent a separate group of information and would be better suited in another table. Additionally, assign a primary key to the table; the Primary Key is ***LicenceNumber*** because it is unique and does not contain repeating information. Now the columns for the table would look something like the following;

LicenceNumber	NumberOfHireDays	DailyRate	Model	Make	Year	DistanceTravelled	YearOfPurchase	InitialCost	Colour

In the second phase of normalization, you need to take another look at your column names to make sure that all columns are dependent on the primary key. This involves eliminating columns that may be partially in the same group, but not totally dependent on the primary key.

Since the CustomerFName and CustomerLName are concerned with a customer's rentals as opposed to the actual car rental information, they are not dependent on the primary key. These columns should be removed and placed in another table, one named ***customers***. Since these fields contain repeating information a new primary key is created ***CustomerID***

CustomerID	CustomerFName	CustomerLName	CompanyName	ContactNumber

Now the original table should contain the following columns: LicenceNumber, NumberOfHireDays, DailyRate, Model, Make, Year, DistanceTravelled, YearOfPurchase, InitialCost and Colour

In the third phase of normalization, you need to re-examine your columns to make sure each column is dependent on the primary key. Consider creating additional tables to eliminate non-dependent primary key columns, if necessary.

LicenceNumber	Model	Make	Year	DistanceTravelled	YearOfPurchase	InitialCost	Colour

The non-dependant fields are added to a new table called ***rentals***, which has no primary key. A new Primary Key is added, called ***RentalID***.

RentalID	NumberOfHireDays	DailyRate

Since the table contains columns that are all dependent upon the primary key, there is no need to further alter the columns for the customer table. Once all of your tables are normalized, you can begin to link tables by assigning foreign keys to your tables. The three tables can be linked by adding two foreign keys in the ***rentals*** table which then joins the two parent tables. A foreign key must have the same data type and size as it matching primary key. The final tables are,

LicenceNumber	Model	Make	Year	DistanceTravelled	YearOfPurchase	InitialCost	Colour

RentalID	NumberOfHireDays	DailyRate	FK_CustomerID	FK_LicenceNumber

CustomerID	CustomerFName	CustomerLName	CompanyName	ContactNumber

Query Terminologies

In the SQL environment the following terminologies are used to describe various aspects of query based languages. There are different version of SQL, however these differences are minor and will not affect the majority of the queries used by modern database systems.

Query Definitions

Clause	A segment of an SQL statement that assists in the selection and manipulation of data.
Keywords	Reserved words used within SQL statements.
Query	A question or command posed to the database.
Statements	Keywords combined with data to form a database query.
Structured Query Language (SQL)	A nonprocedural database programming language used within DBMSs to create, manage, and secure relational databases.

In general terms, query languages can be classified according to whether they are database query languages or information retrieval query languages. The difference is that a database query language attempts to give factual answers to factual questions, while an information retrieval query language attempts to find documents containing information that is relevant to an area of inquiry.

The choice of query language will depend on the data source; in this document we will be using structured query language (SQL) that is suitable for relational database systems. SQL is a standard interactive and programming language for getting information from and updating a database. Although SQL is both an ANSI and an ISO standard, many database products support SQL with proprietary extensions to the standard language.

An information retrieval (IR) query language is a query language used to make queries into database, where the semantics of the query are defined not by a precise rendering of a formal syntax, but by an interpretation of the most suitable results of the query.

Of importance in IR query languages is weighting and ranking, "relevance-orientation", semantic relativism and logic-based probabilism. An example of an IR query language is contextual query language (CQL), a formal language for representing queries to information retrieval systems such as web indexes, bibliographic catalogues and museum collection information.

END of TOPIC ONE