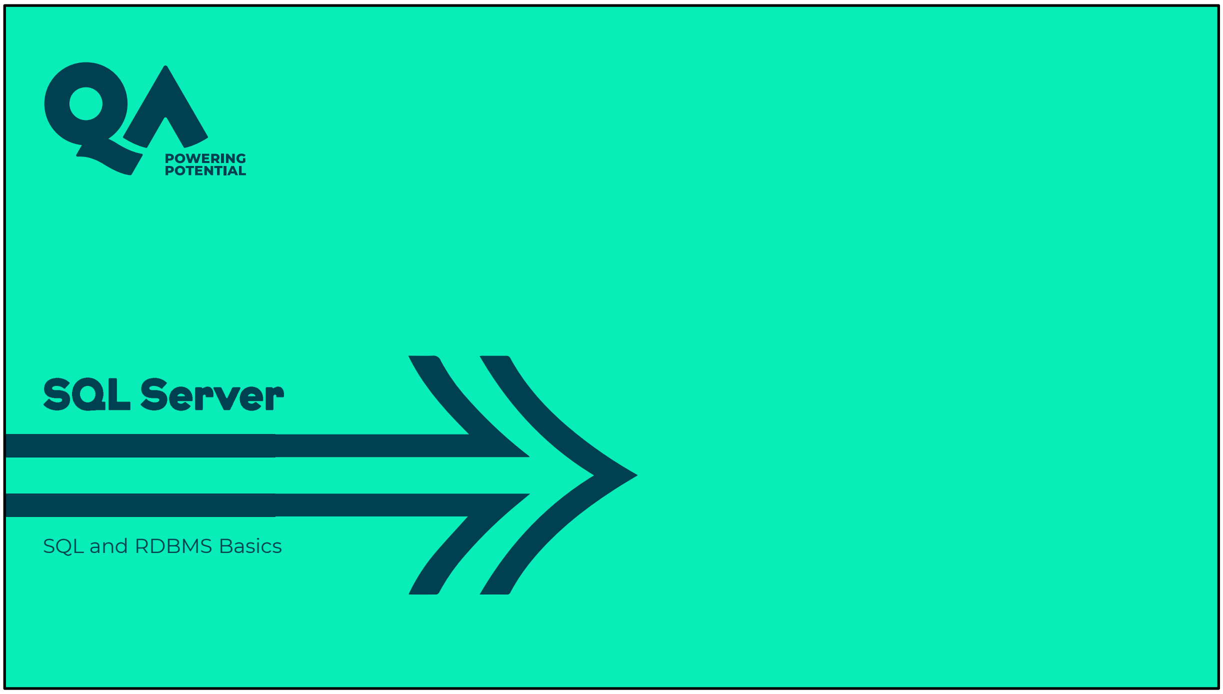


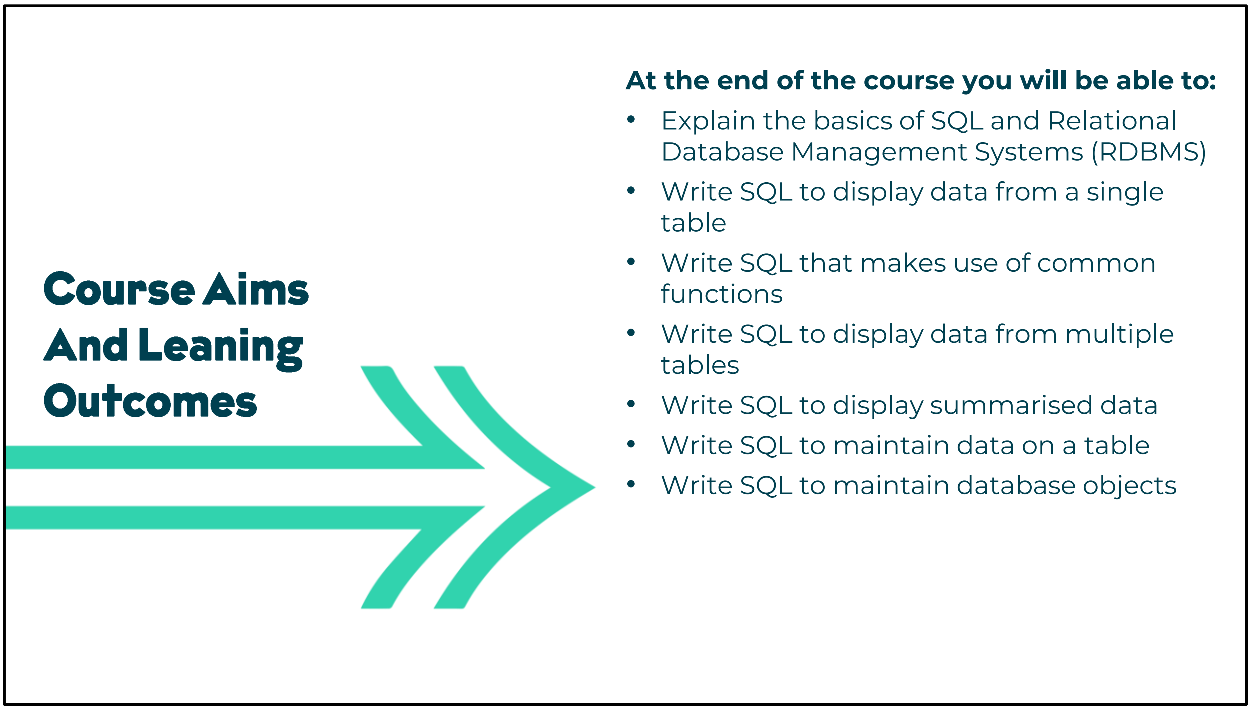
SQL Server

SQL and RDBMS Basics

This module will enable the learner to explain the basics of Structured Query Language (SQL) and Relational Database Management Systems (RDBMS). Specifically the learner will be able to:

- 1.1 Describe the key features and history of SQL
- 1.2 Describe the key features of a Relational Database Management System (RDBMS)
- 1.3 List and describe the high level categories used to group SQL commands
- 1.4 Install MySQL / SQL Server and setup the **qastore** database

Code examples will be used to explain and demonstrate the content and for leaners to imitate. Supporting exercises will provide practice opportunities. At the end of the module there is a short quiz to review the topic.



Course Aims And Leaning Outcomes

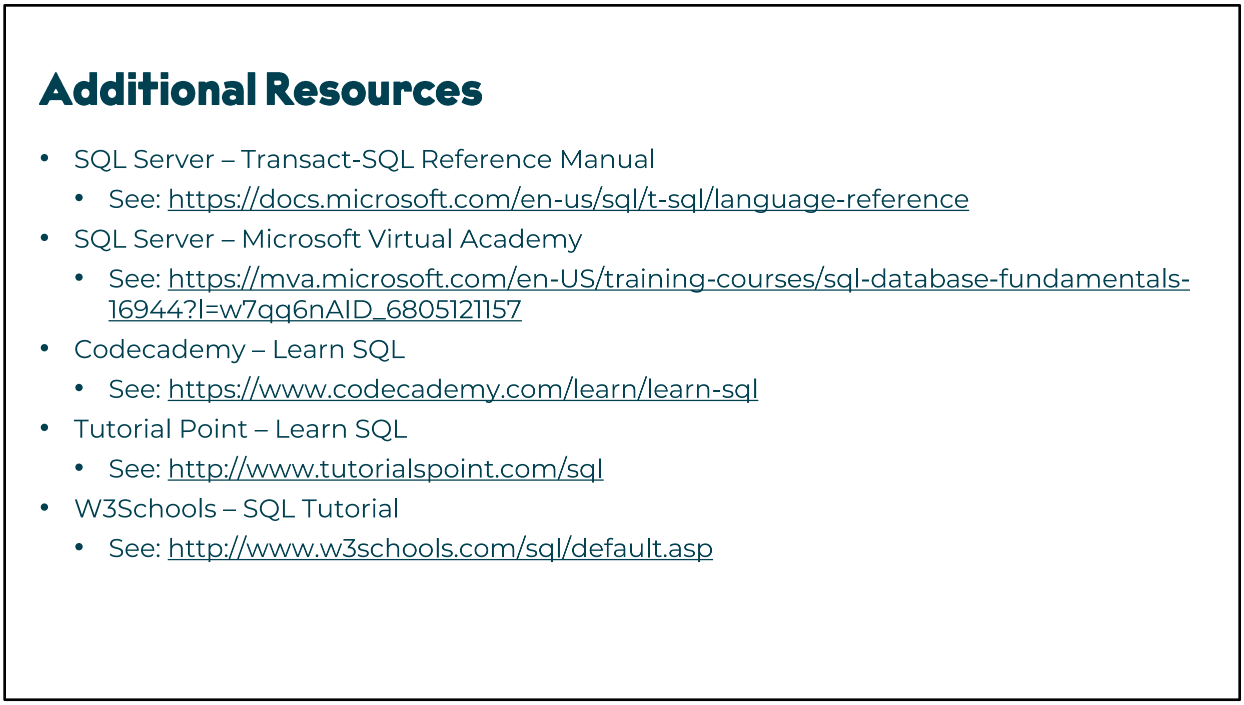
At the end of the course you will be able to:

- Explain the basics of SQL and Relational Database Management Systems (RDBMS)
- Write SQL to display data from a single table
- Write SQL that makes use of common functions
- Write SQL to display data from multiple tables
- Write SQL to display summarised data
- Write SQL to maintain data on a table
- Write SQL to maintain database objects

The aim of this course is to enable learners to develop the skills required to create queries, provide reports and manipulate data in a Relational Database Management System (RDBMS).

At the end of the course the learner will be able to:

- 1. Explain the basics of SQL and Relational Database Management Systems (RDBMS)
- 2. Write SQL to display data from a single table
- 3. Write SQL that makes use of common functions
- 4. Write SQL to display data from multiple tables
- 5. Write SQL to display summarised data
- 6. Write SQL to display data for more complex requests
- 7. Write SQL to maintain data on a table
- 8. Write SQL to maintain database objects

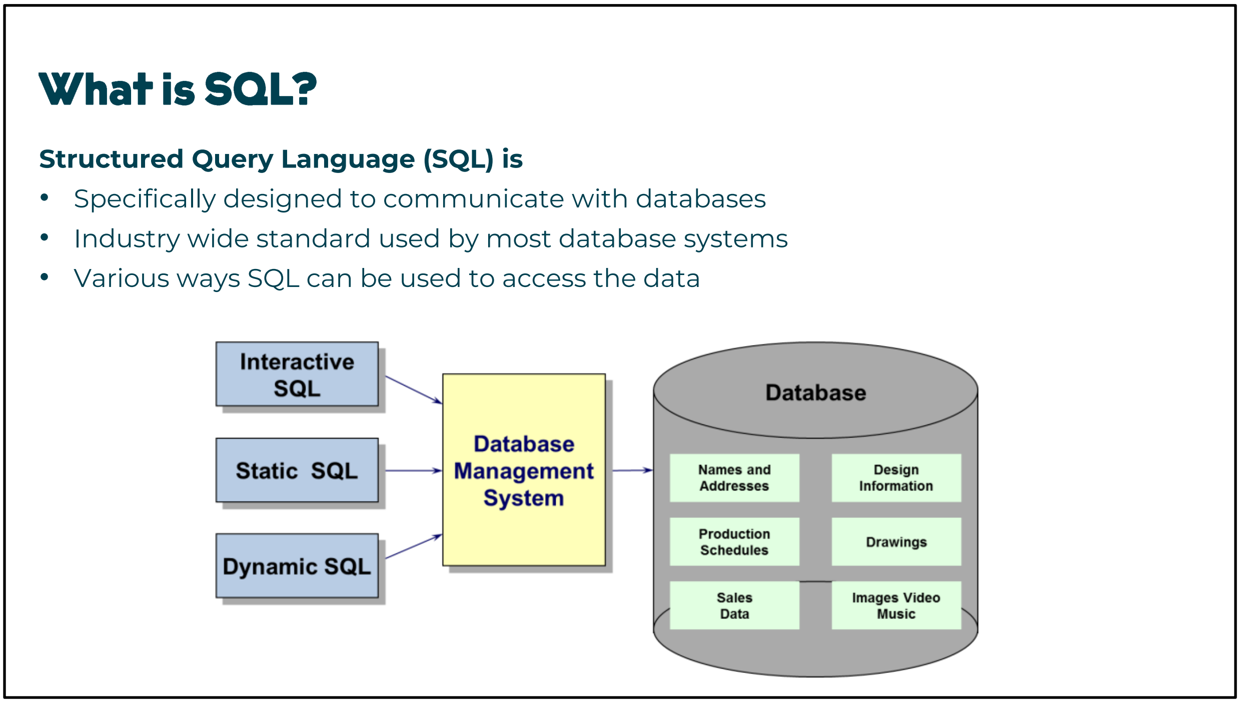


Additional Resources

- SQL Server – Transact-SQL Reference Manual
  - See: https://docs.microsoft.com/en-us/sql/t-sql/language-reference
- SQL Server – Microsoft Virtual Academy
  - See: https://mva.microsoft.com/en-US/training-courses/sql-database-fundamentals-16944?l=w7qq6nAID_6805121157
- Codecademy – Learn SQL
  - See: https://www.codecademy.com/learn/learn-sql
- Tutorial Point – Learn SQL
  - See: http://www.tutorialspoint.com/sql
- W3Schools – SQL Tutorial
  - See: http://www.w3schools.com/sql/default.asp

There are numerous resources on the internet on SQL Server.
Some recommended additional resources are listed above with their URLs.

- SQL Server – Transact-SQL Reference Manual: Comprehensive online documentation.
- SQL Server – Microsoft Virtual Academy: Range of online courses for developers, database administrators and programming interfaces.
- Codecademy – Learn SQL: Online course with practice projects.
- Tutorial Point – Learn SQL: Online course with built in practice environment.
- W3Schools – SQL Tutorial: Online course with built in practice environment.



What is SQL?

Structured Query Language (SQL) is

- Specifically designed to communicate with databases
- Industry wide standard used by most database systems
- Various ways SQL can be used to access the data

Structured Query Language (SQL) is a programming language specifically designed to communicate with databases. It is an industry wide standard used by most database systems.

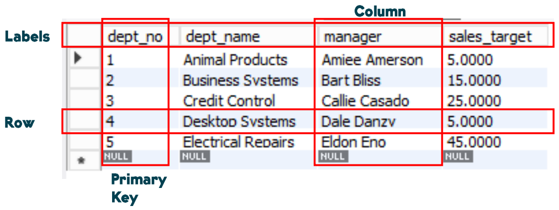
SQL can be used in various ways to access and perform actions on the database and data:

- Interactive SQL – Direct Input using the SQL Client application of DBMS package such as MySQL Workbench. SQL statements are entered within the application, executed and results output without having to write an application in any sort of structured language.
- Static SQL – The developer embeds SQL that runs repeatedly in a host language. There are several options to do this:
  - Wrap the SQL inside keywords like EXEC SQL and END EXEC – For example COBOL.
  - Pass SQL statements as arguments to functions to communicate with the database via intermediate software driver or Application Programming Interface (API) such as:
    - Open Database Connectivity (ODBC) for Windows.
    - Java Database Connectivity (JDBC) for Java.
  - Use an Integrated Development Environment (IDE).
- Dynamic SQL – The developer writes code to create SQL statements whose content cannot be predicted at development time. Examples are script or programs on web servers that dynamically communicate with the database using languages such as PERL or PHP.

Database Table

A database is simply a means to store data in a structured manner

- Composed of one more tables
- Data organised into rows and columns
  - Labels identify the data in each column
  - Each row of data also known as a record
  - Data not stored in any specific order – Sort as needed when
  - Primary Key used to uniquely identify each record
- Easy to reference and maintain



dept_no	dept_name	manager	sales_target
1	Animal Products	Amiee Amerson	5.0000
2	Business Systems	Bart Bliss	15.0000
3	Credit Control	Callie Casado	25.0000
4	Desktop Systems	Dale Danzy	5.0000
5	Electrical Repairs	Eldon Eno	45.0000
NULL	NULL	NULL	NULL

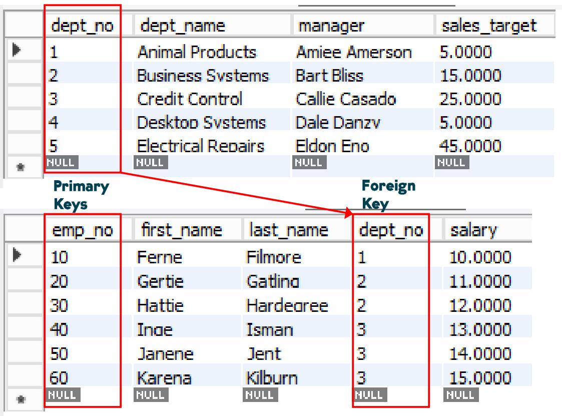
A database is simply a means to store data in a structured manner. It is composed of one more tables with its data organised into rows and columns. Labels are used to identify the data in each column. Each row of data is also known as a record. The data is not stored in any specific order; instead it is sorted and ordered as needed when it is retrieved.

A Primary Key is used to uniquely identify each record.

The structure is easy to reference and maintain.

Multiple Tables

- **Tables related through Primary / Foreign Key relationship**

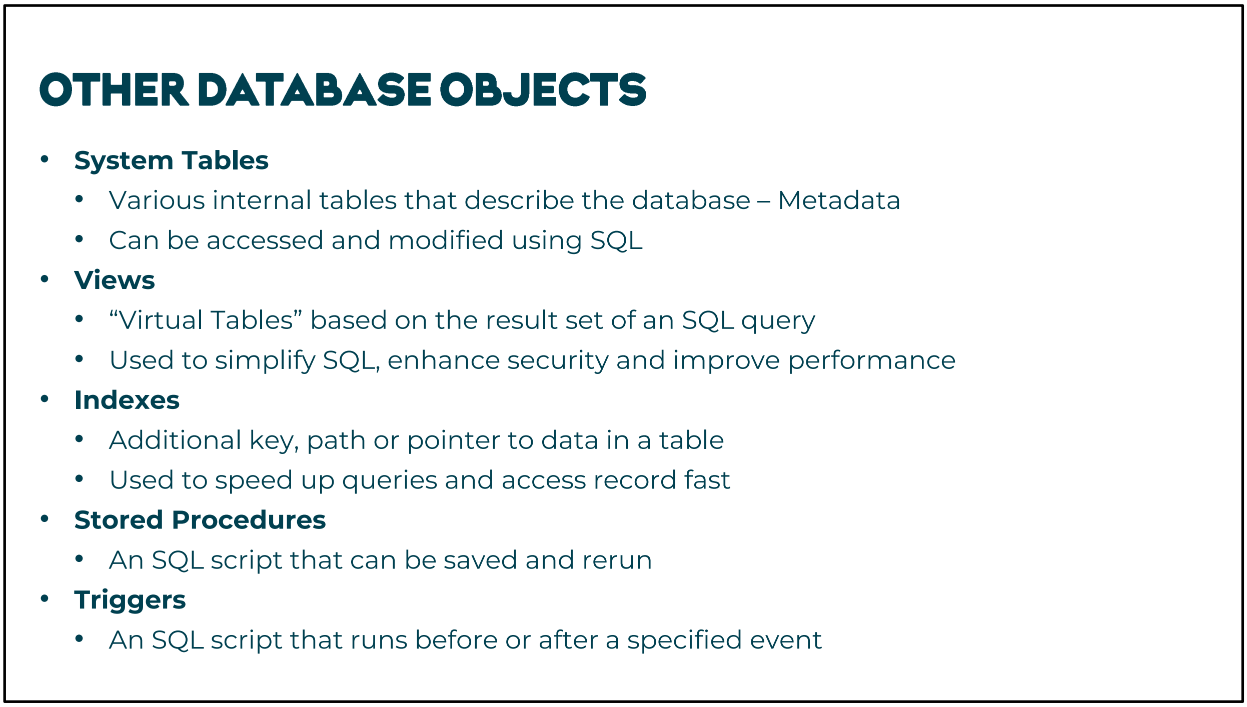


dept_no	dept_name	manager	sales_target
1	Animal Products	Amiee Amerson	5.0000
2	Business Systems	Bart Bliss	15.0000
3	Credit Control	Callie Casado	25.0000
4	Desktop Systems	Dale Danzy	5.0000
5	Electrical Repairs	Eldon Eno	45.0000
NULL	NULL	NULL	NULL

emp_no	first_name	last_name	dept_no	salary
10	Ferne	Filmore	1	10.0000
20	Gertie	Gatling	2	11.0000
30	Hattie	Hardegree	2	12.0000
40	Inge	Isman	3	13.0000
50	Janene	Jent	3	14.0000
60	Karena	Kilburn	3	15.0000
NULL	NULL	NULL	NULL	NULL

A relational database will typically consist of multiple tables linked or related by corresponding Primary and Foreign Keys. The rules or relationships will be specified when the database is set up and enforce the Referential Integrity of the database.

In the above example **dept_no** is the Primary Key on the **dept** table and the Foreign Key on the **salesperson** table. This means that a salesperson record can not be added to the **salesperson** table with a given **dept_no** if there is not a department record on the **dept** table with that **dept_no** as its Primary Key. Also a record on the **dept** table can not be deleted if its Primary Key or **dept_no** is currently allocated to one or more salesperson records on the **salesperson** table.

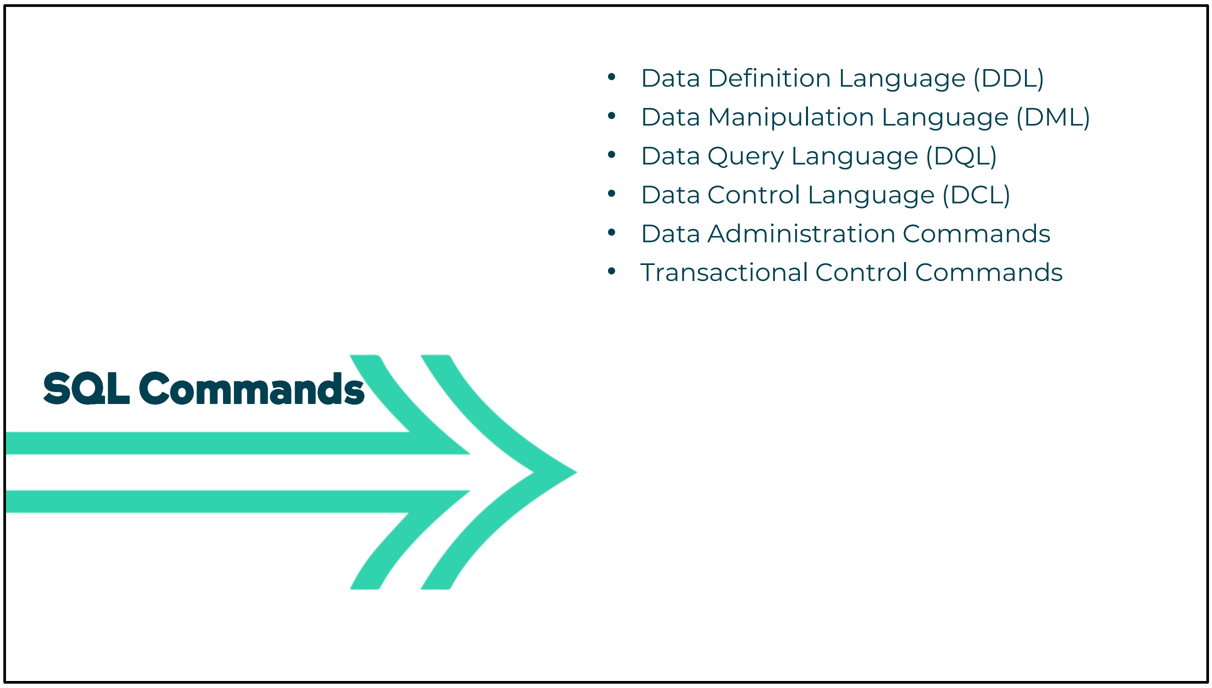


OTHER DATABASE OBJECTS

- **System Tables**
  - Various internal tables that describe the database – Metadata
  - Can be accessed and modified using SQL
- **Views**
  - “Virtual Tables” based on the result set of an SQL query
  - Used to simplify SQL, enhance security and improve performance
- **Indexes**
  - Additional key, path or pointer to data in a table
  - Used to speed up queries and access record fast
- **Stored Procedures**
  - An SQL script that can be saved and rerun
- **Triggers**
  - An SQL script that runs before or after a specified event

In addition to Tables, Columns, Row, Primary and Foreign Keys there a number of other database objects:

- System Tables - Various internal tables that describe the database also know as “metadata” or “data about data”. System tables can be accessed and modified using SQL just like normal tables.
- Views – A “Virtual Table” based on the result set of an SQL query. They are used to simplify complex or repeated SQL, enhance security and improve performance.
- Indexes – These are additional keys, paths or pointers to data in a table and are used to speed up queries and access record more quickly.
- Stored Procedures – This is an SQL script that can be saved and rerun avoiding the need to recode SQL that is used on a regular basis.
- Triggers – This is an SQL script that runs before or after a specified event saving the time to code the SQL and the possibility that it may not be done.

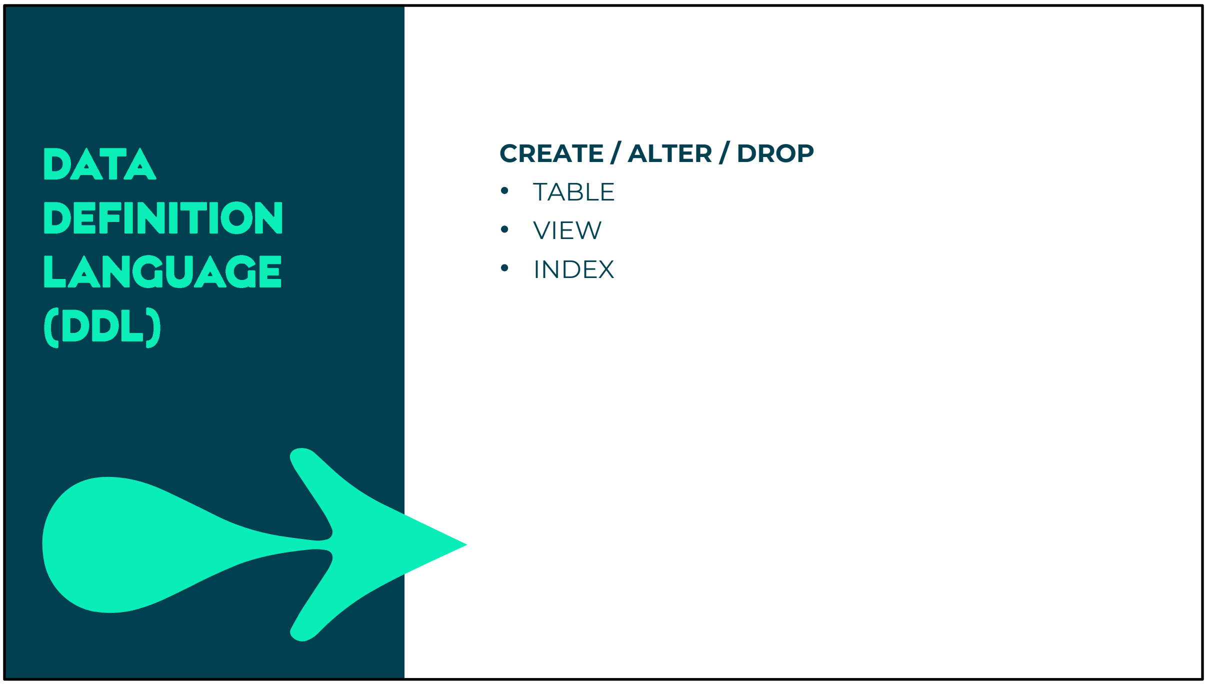


SQL Commands

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Query Language (DQL)
- Data Control Language (DCL)
- Data Administration Commands
- Transactional Control Commands

There are several types or categories of SQL commands to perform various functions such as building databases, manipulating objects, adding, updating and deleting data, performing queries, managing database access and database administration.
The main types are:

- Data Definition Language (DDL) – Used to maintain database objects such as a table.
- Data Manipulation Language (DML) – Used to maintain data within a database object like a table.
- Data Query Language (DQL) – Used to query the database and display data.
- Data Control Language (DCL) – Used to control access to the database.
- Data Administration Commands – Used to perform audits and analyse operations on the database.
- Transactional Control Commands – Used to manage database transactions.



DATA DEFINITION LANGUAGE (DDL)

CREATE / ALTER / DROP

- TABLE
- VIEW
- INDEX

Data Definition Language (DDL) SQL commands are used to maintain database objects such as a table, view or index:

- CREATE is the command used to add an object.
- ALTER to update an object.
- DROP to delete an object.

Common objects are:

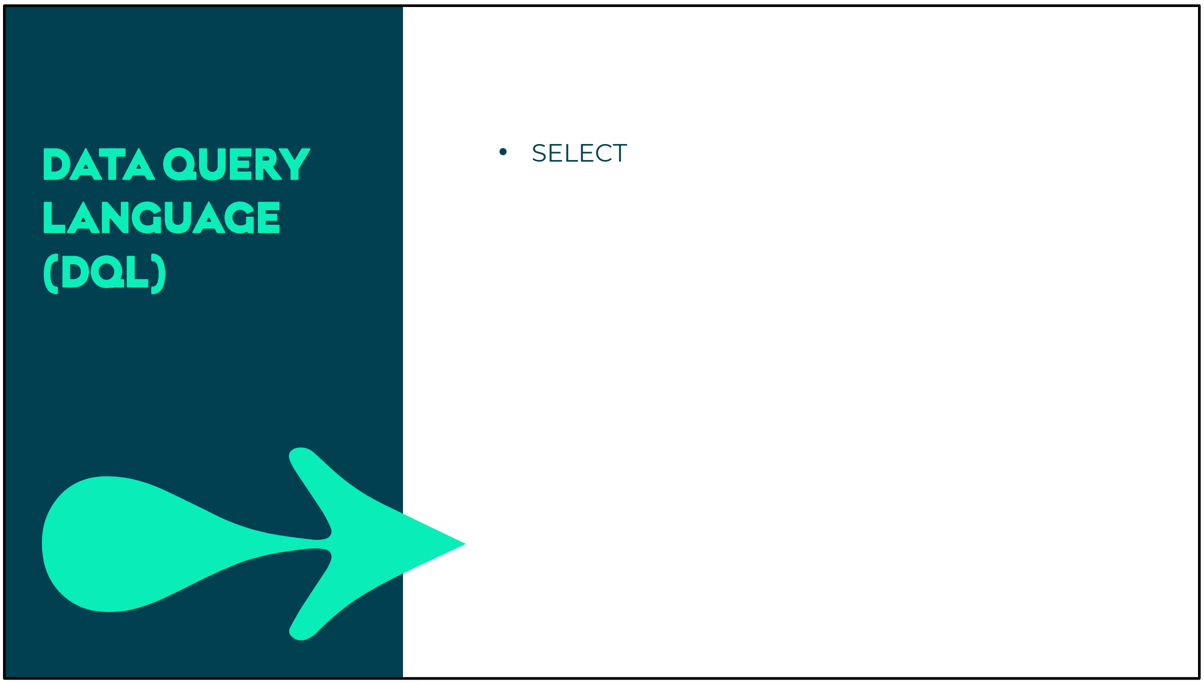
- TABLE – Primary storage object for data in a relational database; in its simplest form it consists of row or records and columns or fields.
- VIEW – A virtual table that looks and acts like a table but does not require any physical storage.
- INDEX – A pointer to data in a table like an index in a book.

Data Manipulation Language (DML)

- INSERT
- UPDATE
- DELETE

Data Manipulation Language (DML) SQL commands are used to maintain data within a database object like a table:

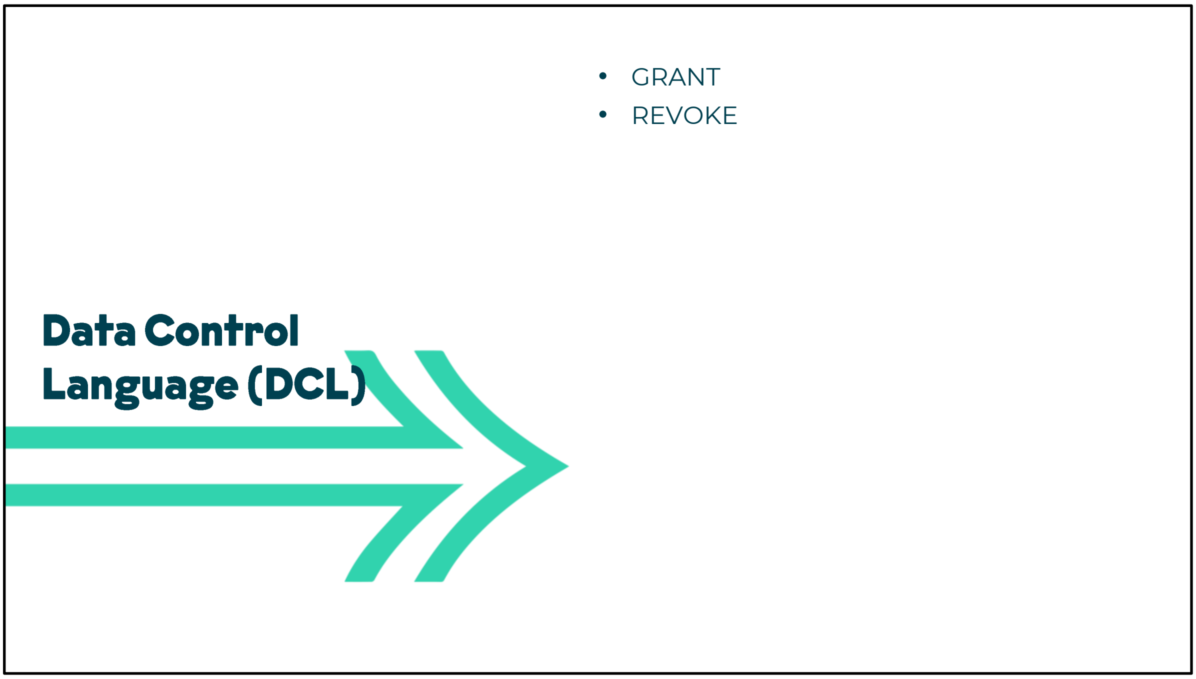
- INSERT – Adds a new record to a database table.
- UPDATE – Amends an existing record on a database table.
- DELETE – Removed an existing record on a database table.



DATA QUERY LANGUAGE (DQL)

- SELECT

Data Query Language (DQL) SQL commands are used to query the database and display data.
SELECT is the base command with many options and clauses to define which tables and columns to use, the conditions for selection and the order the data will be displayed in.

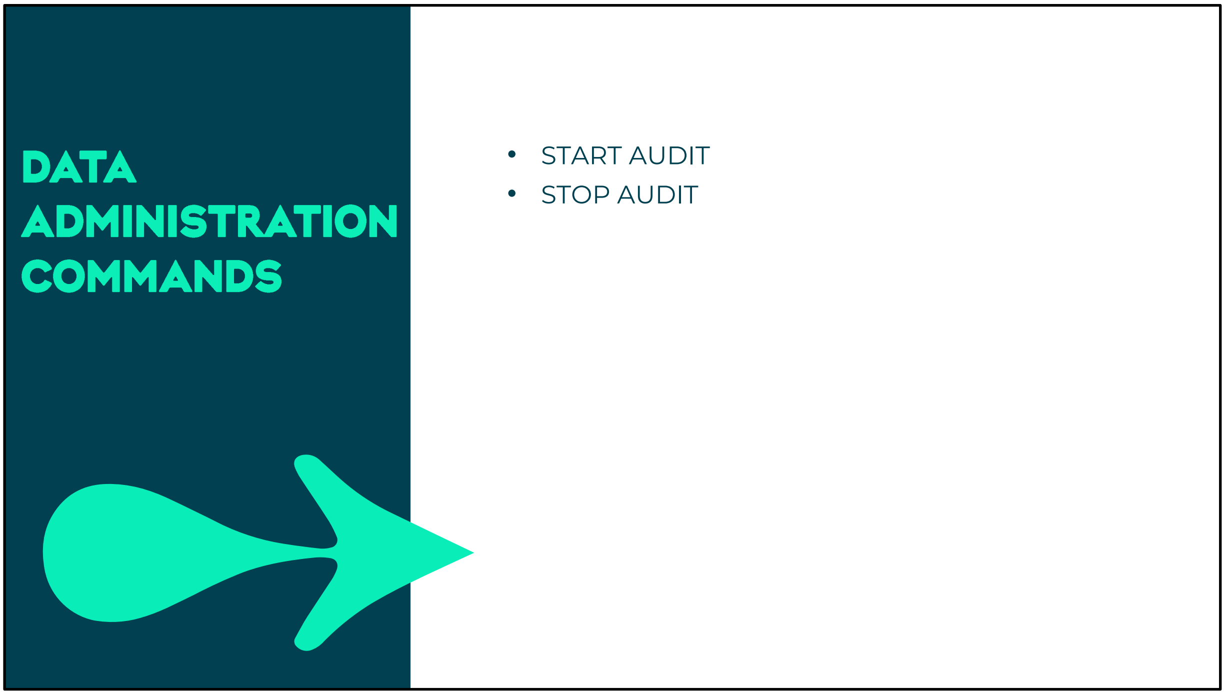


Data Control Language (DCL)

- GRANT
- REVOKE

Data Control Language (DCL) SQL commands are used to control access to the database:

- GRANT – Used to give privileges to access specific database objects and perform specific tasks.
- REVOKE – Used to remove or modify privileges previously granted.



DATA ADMINISTRATION COMMANDS

- START AUDIT
- STOP AUDIT

Data Administration SQL commands are used to perform audits and analyse operations on the database including system performance.
Two general data administration commands are:

- START AUDIT
- STOP AUDIT

Transactional Control Commands

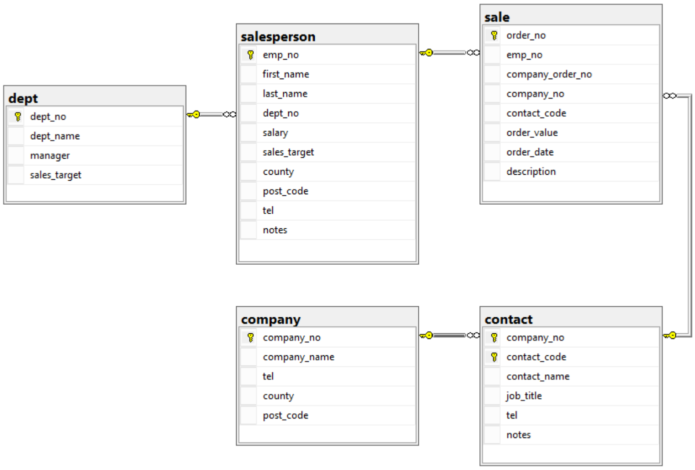
- COMMIT
- ROLLBACK
- SAVEPOINT
- SET TRANSACTION

Transactional Control SQL commands are used to manage database transactions:

- COMMIT – Saves database transactions.
- ROLLBACK – Undoes database transactions.
- SAVEPOINT – Create a point within a database transactions to potentially ROLLBACK to.
- SET TRANSACTION – Gives a transaction a name.

Most SQL products, MySQL included, have an option to automatically commit any changes to data.

qastore Database



dept
- dept_no
- dept_name
- manager
- sales_target

salesperson
- emp_no
- first_name
- last_name
- dept_no
- salary
- sales_target
- county
- post_code
- tel
- notes

sale
- order_no
- emp_no
- company_order_no
- company_no
- contact_code
- order_value
- order_date
- description

company
- company_no
- company_name
- tel
- county
- post_code

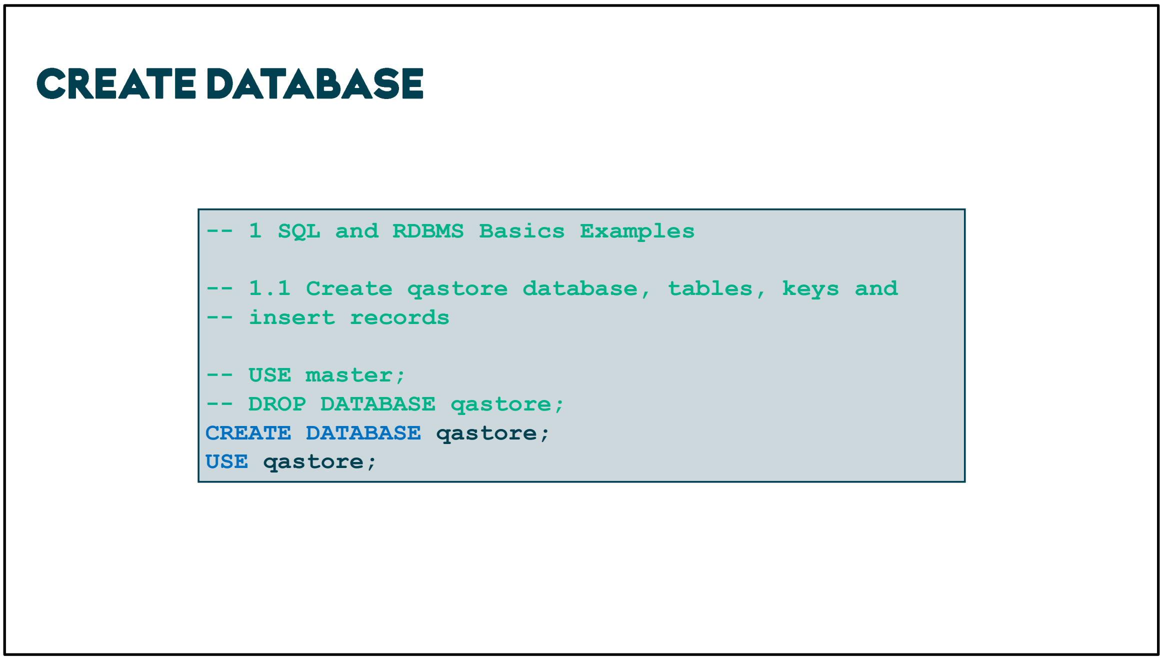
contact
- company_no
- contact_code
- contact_name
- job_title
- tel
- notes

qastore is the database we will be using for all the examples and exercises on the course.

It is a simple Sales system with five interconnecting tables:

- dept – Is a list of departments within the sales organisation.
- salesperson – The sales staff in the organisation who are assigned to a department
- sale – The sales or orders that each salesperson has made.
- contact – The person from an external organisation that each sale was made to.
- company – The external organisation that the contact belongs to.

This is an example of a Data Model that you covered in the Data Modelling course. Each of the five boxes is a table or Entity on the Data Model with its corresponding Attributes within the box. The Primary Key is indicated by a yellow key next to appropriate Attribute and the Foreign Key by a red diamond. For all Attributes their name, data type and size, where applicable is given. Finally the lines between the tables specify the relationships between the tables. This is referential integrity between each table as defined by the Primary and Foreign Keys. For example a salesperson can only be assigned to a valid dept_no that is on the dept table. Also we will not be able to delete a dept that currently has a salesperson assigned to its dept_no.



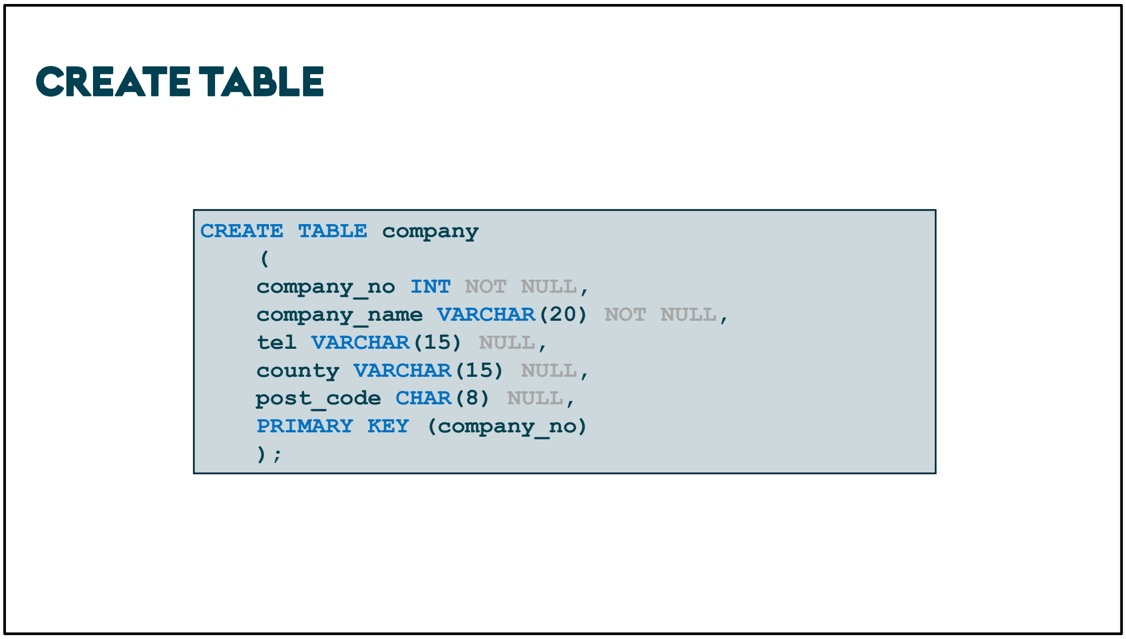
CREATE DATABASE

```
-- 1 SQL and RDBMS Basics Examples

-- 1.1 Create qastore database, tables, keys and
-- insert records

-- USE master;
-- DROP DATABASE qastore;
CREATE DATABASE qastore;
USE qastore;
```

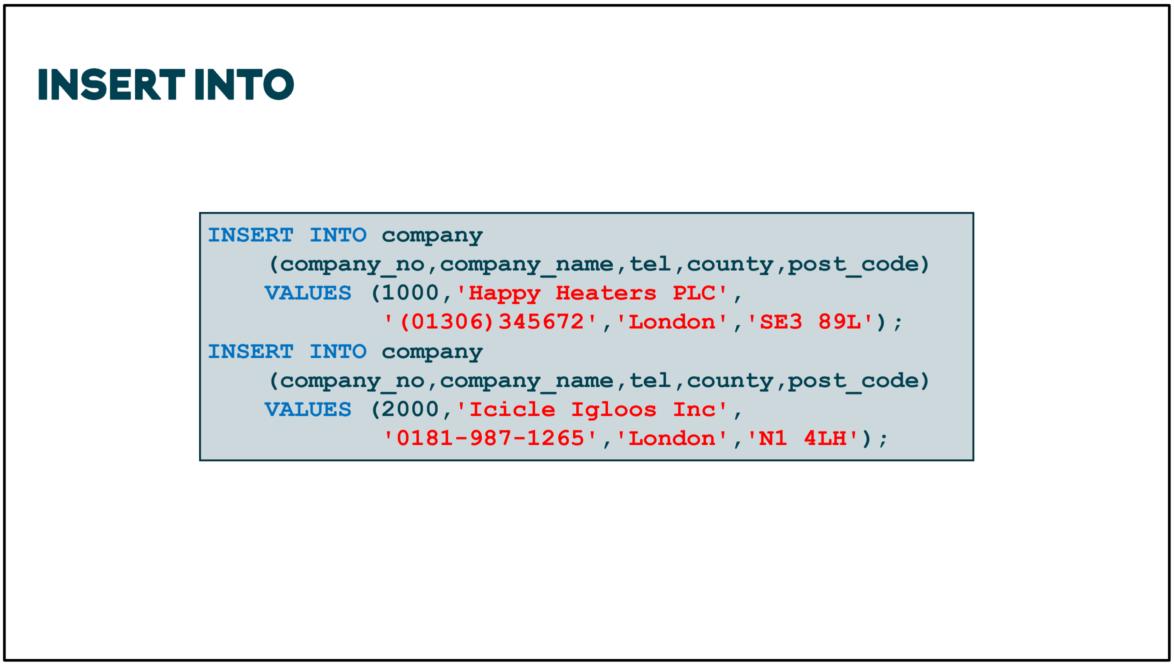
This is the first part of the SQL to add the qastore database.
The first few lines are comments describing the SQL.
The USE command specifies the use of the master database to setup the database required.
The DROP command will delete the database if it already exists as it is not possible to have two databases with the same name.
The CREATE command will add a new database.
The USE command will the specify that this is the database we want to use for subsequent SQL.



CREATE TABLE

```
CREATE TABLE company
    (
    company_no INT NOT NULL,
    company_name VARCHAR(20) NOT NULL,
    tel VARCHAR(15) NULL,
    county VARCHAR(15) NULL,
    post_code CHAR(8) NULL,
    PRIMARY KEY (company_no)
    );
```

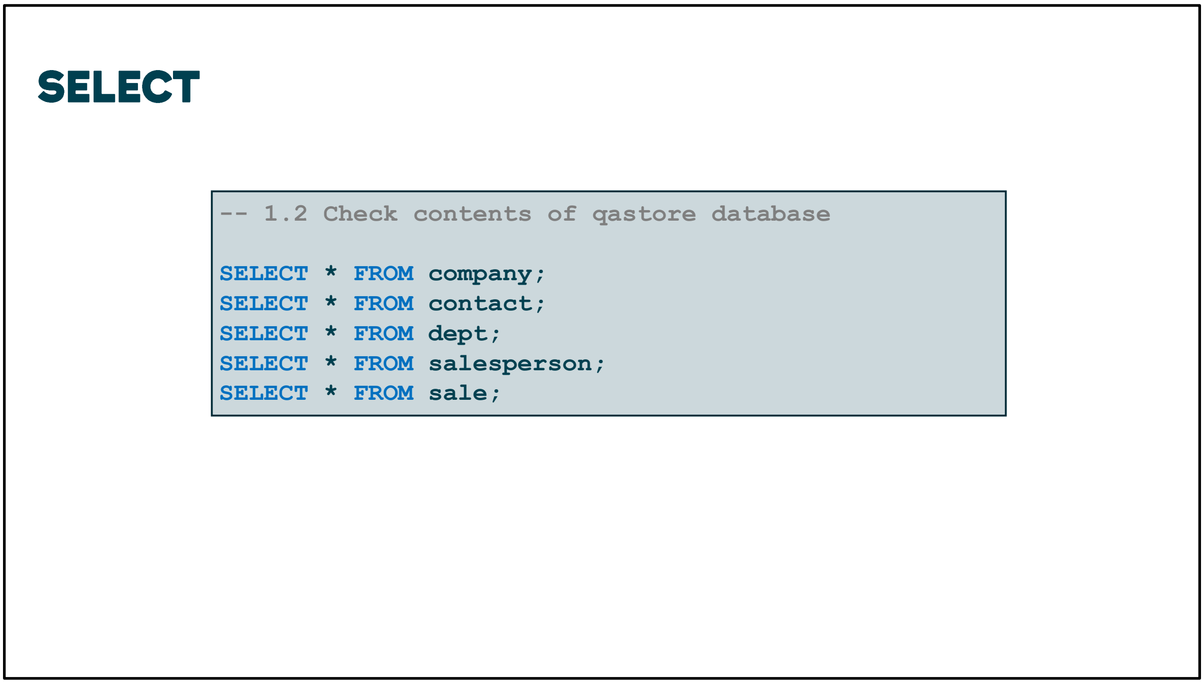
Next the SQL adds the tables for the qastore database.
The CREATE TABLE command specifies the name of the table and the columns or fields that make up the table. Each column is given a name or label, the type of data stored in the column and whether it is mandatory (NOT NULL) or optional (NULL) for data to entered in the column when a record is created.
Finally the PRIMARY KEY command specifies the column or columns that will make the Primary Key or unique identifier for each record on the table.



INSERT INTO

```
INSERT INTO company
    (company_no,company_name,tel,county,post_code)
    VALUES (1000,'Happy Heaters PLC',
            '(01306)345672','London','SE3 89L');
INSERT INTO company
    (company_no,company_name,tel,county,post_code)
    VALUES (2000,'Icicle Igloos Inc',
            '0181-987-1265','London','N1 4LH');
```

Next the SQL adds the data or records for each table.
The INSERT INTO command specifies the table the data will be added to and the columns that will be updated. In this case all the columns for each table in the order specified on the database.
The VALUES command specify the actual data values that will update each column in sequence. Note the colour of data and use of quotations to indicate numerical and text data.

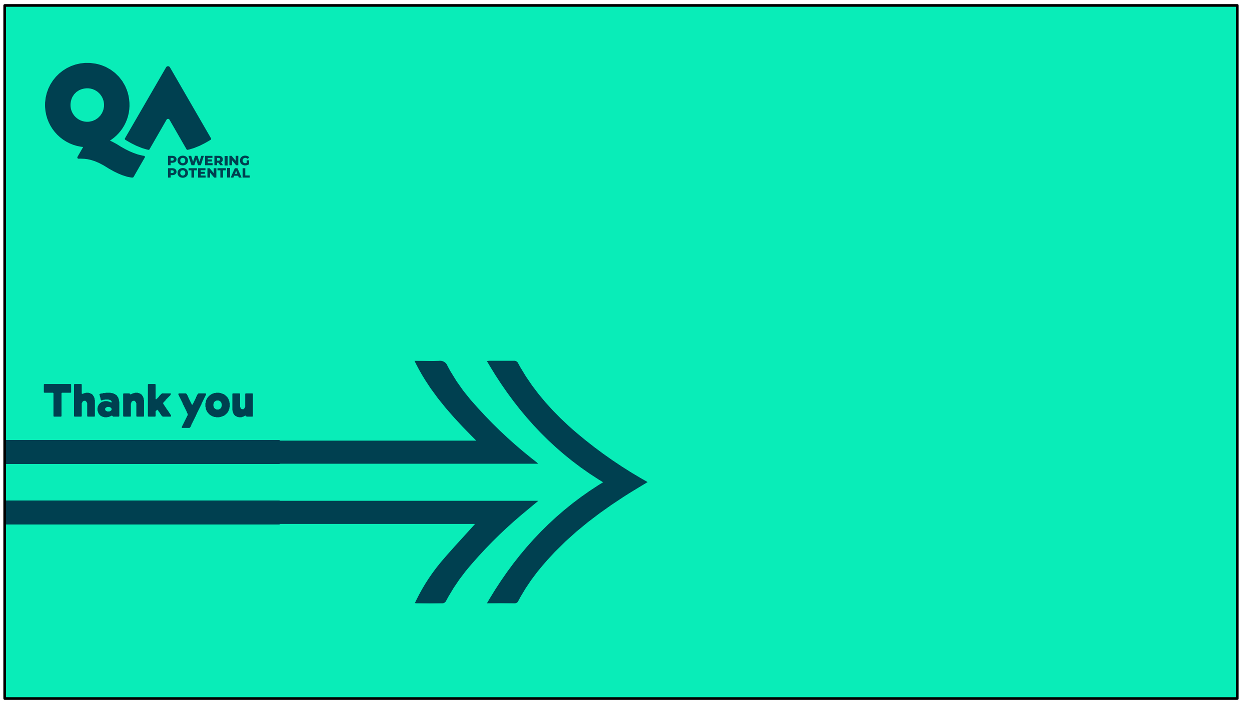


SELECT

```
-- 1.2 Check contents of qastore database

SELECT * FROM company;
SELECT * FROM contact;
SELECT * FROM dept;
SELECT * FROM salesperson;
SELECT * FROM sale;
```

Finally the SQL checks the records added to each table.
The SELECT command retrieves the records from the specified table. The “*” is shorthand or a wildcard for all the columns in the table.



POWERING
POTENTIAL

Thank you