



1.3.2 SQL

CREATE TABLE

The CREATE TABLE command is used to **define a table**, its attributes, the datatypes of the attributes and declare the primary key.

Let's say we needed to create a table to store engineers working for a factory. It is required that we store their ID number, surname, initial and phone extension.

```
CREATE TABLE Engineer (  
    EngineerID INTEGER PRIMARY KEY,  
    Surname VARCHAR(16),  
    Initial CHAR(1),  
    PhoneExtension INTEGER  
);
```

Engineer

EngineerID	Surname	Initial	PhoneExtension

This creates an empty Engineer table with the EngineerID fields as the primary key.

VARCHAR(16) means up to 16 characters.

CHAR(1) means 1 character.

Other data types include DATE, DECIMAL and BOOLEAN.





UPDATE

The UPDATE command is used to **modify an existing record** in the tables of the database.

Let's change the phone extension of engineer 228 of the Engineer table.

```
UPDATE Engineer  
SET PhoneExtension = 123  
WHERE EngineerID = 228;
```

Engineer

EngineerID	Surname	Initial	PhoneExtension
182	Hughes	P	387
228	Sanderson	T	459 123

The **WHERE** clause specifies which records are modified.

Multiple conditions can be combined using AND and OR.





DELETE

The DELETE command is used to **remove a record** from the table.

Let's remove the engineer with EngineerID 228 from the Engineer table.

```
DELETE FROM Engineer  
WHERE EngineerID = 228;
```

Engineer

EngineerID	Surname	Initial	PhoneExtension
182	Hughes	P	387
228	Sanderson	T	123

Be careful to include the WHERE condition or all records in the table will be deleted!

DROP TABLE

The DROP TABLE command **permanently deletes a table and all of its data**.

```
DROP TABLE Engineer;
```





SELECT

The SELECT command is used to **query** a database to extract information.

Let's get the engineer ID and surname of all engineers that have the initial P.

```
SELECT EngineerID, Surname  
FROM Engineer  
WHERE Initial = 'P';
```

Engineer

EngineerID	Surname	Initial	PhoneExtension
182	Hughes	P	387
228	Sanderson	T	123
306	Wong	P	512

This would output (182, 'Hughes'), (306, 'Wong').

The **ORDER BY** command is used to sort the output by a specific field.

For example, ORDER BY Surname;





Multiple Table Queries

If a query requires data from multiple tables, the tables can be joined using a primary key–foreign key relationship. The fields used to join the tables are specified in the ON clause.

Let's output all dates and locations for all problems assigned to the engineer with surname Hughes and initial P.

```
SELECT Date, Location
FROM PROBLEM
INNER JOIN Engineer
ON Problem.EngineerID = Engineer.EngineerID
WHERE Surname = 'Hughes' AND Initial = 'P';
```

Engineer

EngineerID	Surname	Initial	PhoneExtension
182	Hughes	P	387
228	Sanderson	T	123
306	Wong	P	512

Problem

IncidentID	Date	Location	ProblemDescription	EngineerID
1866	02-May-25	Room A6	Printer not working	182
1867	03-May-25	Workshop 4	Hard drive fault	228
1868	04-May-25	Room B22	Keyboard faulty	182
1869	04-May-25	Warehouse 3	Computer overheating	306

This would output: ('02-May-25', 'Room A6') , ('04-May-25', 'Room B22')

