SQL – Structured Query Language

Summary

SQL is a language used in databases to define data models / data structures based on tables and relations. It is also used, to query and modify (CREATE, UPDATE, DELETE) data, that is based on these data structures.

Filter data, sort data

modelled by

tables related to each other

INSERT

Defining tables

```
CREATE TABLE "List" (
        "Id"     INTEGER NOT NULL,
        "Title"  TEXT NOT NULL,
        "Description" TEXT,
        PRIMARY KEY("Id" AUTOINCREMENT)
);
CREATE TABLE "ToDo" (
        "Id"     INTEGER NOT NULL,
        "Title"  TEXT NOT NULL,
        "Description" TEXT,
        "ListId"         INTEGER NOT NULL,
        FOREIGN KEY("ListId") REFERENCES List(Id),
        PRIMARY KEY("Id" AUTOINCREMENT)
);
```

Table will hold all the todo lists

Primary Key is like an identifier used to find entries in the table

Querying examples

SELECT * FROM List;

SELECT Name FROM List;

SELECT * FROM List where Title = 'Cleaning'

SELECT * FROM List where Title = 'Cleaning' and Description is null

SELECT * FROM List where Description is not null

SELECT * FROM List where Title = 'Cleaning' or Description is not null

Wildcards

SELECT * FROM List where Title like 'Clea%'

SELECT * FROM List where Title like '%ning'

SELECT * FROM List where Title like '%lean%'

Insert

INSERT INTO List (Title, Description) VALUES("Shopping", "This list keeps track of clothes to buy");

Update

Update List Set Description = 'Clean the garden and the house' where Id = 1;

DELETE

DELETE FROM LIST where Id = 1;

JOIN

SELECT * FROM List l

INNER JOIN ToDo t ON l.Id = t.ListId;

SELECT * FROM List l

LEFT JOIN ToDo t ON l.Id = t.ListId;

Aggregate

```
SELECT l.Title, COUNT(t.Id) FROM List l
JOIN ToDo t ON l.Id = t.ListId
GROUP BY l.Id
```

13. 3:39 To Prevent redundancy is to normalize the data