

Structured Query Language (SQL)

Chase Yoshida

Here's my problem

iPhone 11	64GB	699
iPhone 11 Pro	64GB	999
iPhone 11 Pro	256GB	1149
iPhone 12	64GB	799
iPhone 12 Pro	128GB	999

Solution: Normalized Tables

FK1_Model	FK2_Storage	
1	1	699
2	1	999
2	3	1149
3	1	799
4	2	999

1	iPhone 11
2	iPhone 11 Pro
3	iPhone 12
4	iPhone 12 Pro

1	64GB
2	128GB
3	256GB

Terminology

- Schema
 - Groups related objects and lets you find the objects by name.
- Table
 - A set of columns and rows

Categories

- DDL - Data Definition Language
 - CREATE TABLE <table name> (<column_name1> PRIMARY KEY, <column_name2>);
- DML - Data Manipulation Language
 - INSERT INTO <table name> ();
- DQL - Data Query Language
 - SELECT FROM <table name> ;
- DCL - Data Control Language
- DBA - Database Administrator

DDL Commands

- CREATE TABLE <tablename> (
 - <columnname1> PRIMARY KEY
 - , <columnname2>, ...
-);
- VARCHAR() - variable character
 - Ex: VARCHAR(5)
 - "Word1"
- NUMERIC(<digits>, <decimal places>)
 - Ex: NUMERIC(2, 2)
 - "19.99"

Example

```
CREATE TABLE iphone (  
    iName VARCHAR(20) PRIMARY KEY  
    , iStorage VARCHAR(5)  
    , iPrice NUMERIC(4, 2)  
);
```

Example cont.

```
INSERT INTO iPhone VALUES
```

```
    ('iPhone 11', '64GB', '699.00')
```

```
    , ('iPhone 11 Pro', '64GB', '999.99')
```

```
    , ('iPhone 11 Pro', '256GB', '1149.00')
```

```
    , ('iPhone 12', '64GB', '799.00')
```

```
    , ('iPhone 12 Pro', '128GB', '999.00')
```

```
;
```