Introduction to SQL

Team JDR

Review

Modifying Tables

Constraints

Join Variants

Set Operations

Indexes and Incrementing

Views

Review

SQL Overview

Clauses

Data Types

Keys

SQL Overview

SQL - Structured Query Language

RDMS - Relational Database Managements System

SQL is used to access, store, and manipulate data that is stored in an RDMS.

SQL Overview - RDMS and Tables

Data in an RDMS is stored in Tables

Tables are composed of entries (rows) and fields (columns)

Example: Table of Students in CS61A

ssid	Major	Age	Name
1	EECS	38	David
2	Geography	39	John
3	EECS	40	Katherine
4	Math	42	Jerry
5	Physics	45	Heather
6	Statistics	47	Sonja
7	EECS	48	George
8	EECS	61	Priscilla

Table

This is the table we will be using for examples:

ssid	Major	Age	Name
1	EECS	38	David
2	Geography	39	John
3	EECS	40	Katherine
4	Math	42	Jerry
5	Physics	45	Heather
6	Statistics	47	Sonja
7	EECS	48	George
8	EECS	61	Priscilla

Clauses - CREATE

```
CREATE TABLE Students(

ssid INT,

Major VARCHAR(25),

Age INT(99),

Name VARCHAR(10),

PRIMARY KEY(ssid));
```

Clauses - SELECT

```
SELECT *

FROM Students
```

ssid	Major	Age	Name
1	EECS	38	David
2	Geography	39	John
3	EECS	40	Katherine
4	Math	42	Jerry
5	Physics	45	Heather
6	Statistics	47	Sonja
7	EECS	48	George
8	EECS	61	Priscilla

Clauses - WHERE

```
SELECT *

FROM Students

WHERE Major = “EECS”
```

ssid	Major	Age	Name
1	EECS	38	David
3	EECS	40	Katherine
7	EECS	48	George
8	EECS	61	Priscilla

Clauses - WHERE

```
SELECT Name, Major

FROM Students

WHERE Age <= 40
```

Name	Major
David	EECS
John	Geography
Katherine	EECS

Clause - ORDER BY

```
SELECT Name, Age

FROM Students

ORDER BY Age
```

Name	Age
David	38
John	39
Katherine	40
Jerry	42
Heather	45
Sonja	47
George	48
Priscilla	61

Clause - GROUP BY

```
SELECT Major, Count(ssid)

FROM Students

GROUP BY Major
```

Major	Count(ssid)
EECS	4
Geography	1
Math	1
Physics	1
Statistics	1

Clause - HAVING

```
SELECT Major, Count(ssid)

FROM Students

GROUP BY Major

HAVING Count(ssid) > 2
```

Major	Count(ssid)
EECS	4

Data Types

CHAR(size)

VARCHAR(size)

INT(size)

BOOL or BOOLEAN

DOUBLE(size, d)

FLOAT(p)

Keys

Primary Keys

Secondary Keys

Modifying Tables

Insert

Update

Delete

Modifying Tables - Insert

Gender	Breed	Age	Name
M	Beagle	3	David
M	Boxer	1	Tyson
F	Corgi	5	Lucky

```
INSERT INTO dogs (Gender, Breed, Age, Name)

VALUES ('F', 'Pomeranian', 8, 'Lucy');
```

Gender	Breed	Age	Name
M	Beagle	3	David
M	Boxer	1	Tyson
F	Corgi	5	Lucky
F	Pomeranian	8	Lucy

Modifying Tables - Update

Gender	Breed	Age	Name
M	Beagle	3	David
M	Boxer	1	Tyson
F	Corgi	5	Lucky
F	Pomeranian	8	Lucy

```
UPDATE dogs
SET Age = 9
WHERE Name='Lucy' AND Breed='Pomeranian';
```

Gender	Breed	Age	Name
M	Beagle	3	David
M	Boxer	1	Tyson
F	Corgi	5	Lucky
F	Pomeranian	9	Lucy

Modifying Tables - Delete

Gender	Breed	Age	Name
M	Beagle	3	David
M	Boxer	1	Tyson
F	Corgi	5	Lucky
F	Pomeranian	9	Lucy

```
DELETE FROM dogs

WHERE Name='Lucy' AND Breed='Pomeranian';
```

Gender	Breed	Age	Name
M	Beagle	3	David
M	Boxer	1	Tyson
F	Corgi	5	Lucky

Constraints

- Not NULL
- Foreign Key
- Unique/Distinct
- Check
- Primary Key
- Default

Constraints - Not NULL

```
CREATE TABLE Politicians(
pol_id INT NOT NULL,
Political_Party VARCHAR(25),
Age INT(99),
Name VARCHAR(10)
);
```

pol_id column will not accept a NULL value

Constraints - Unique/Distinct

```
CREATE TABLE Politicians(

pol_id INT NOT NULL DISTINCT,

Political_Party VARCHAR(25),

Age INT(99),

Name VARCHAR(10)

);
```

pol_id column will not accept a NULL value and will ensure that all values are unique

Constraints - Primary Key

```
CREATE TABLE Politicians(
pol_id INT,
Political_Party VARCHAR(25),
Age INT(99),
Name VARCHAR(10),
PRIMARY KEY (pol_id)
);
```

Primary Keys cannot contain NULL values and they must be unique

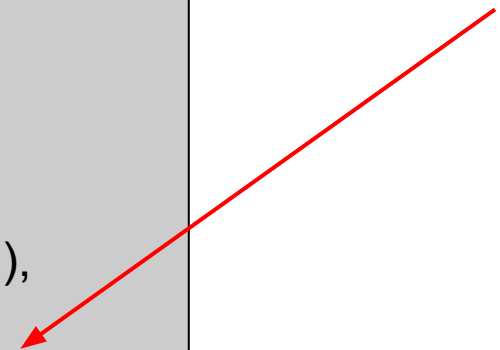
Primary Keys can also consist of multiple columns, but only one Primary Key per table

```
CONSTRAINT PK_pol PRIMARY KEY (pol_id, name)
```

Constraints - Foreign Key

Constraints - Check

```
CREATE TABLE Politicians(
pol_id INT,
Political_Party VARCHAR(25),
Age INT(99),
Name VARCHAR(10),
PRIMARY KEY (pol_id),
CHECK (Age>=25)
);
```



Every value entered for Age must be greater than or equal to 25

Constraints - Default

```
CREATE TABLE Politicians(
pol_id INT,
Political_Party VARCHAR(25) DEFAULT 'Independent',
Age INT(99),
Name VARCHAR(10),
PRIMARY KEY (pol_id),
CHECK (Age>=25)
);
```

The default value for the political party column is Independent