ALLEGHENY
COLLEGE

Keys

Primary Keys

Foreign Keys

Examples and
Code

Introduction to Database Systems: CS312
Foreign Keys and Query Structure

Oliver Bonham-Carter

21 Sept 2020

- **Primary Keys**:
  - Ensures *uniqueness* in a table.
  - All entries in an attribute-primary never repeat
  - Is a unique identifier (i.e., social security number, telephone number, etc)

ALLEGHENY
COLLEGE

Keys

Primary Keys

Foreign Keys

Examples and
Code

What are the types of keys in databases?

- **Foreign Keys**:
  - A constraint to *enforce* the relationships between correlated tables.
  - Allow to create a reference to some piece of information in another table.
  - Foreign key constraints allow checking the referential integrity between tables.
  - Only values that are *supposed* to appear in a particular table are permitted
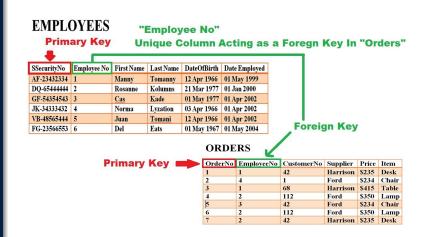
EMPLOYEES

Primary Key

"Employee No"
Unique Column Acting as a Foregn Key In "Orders"

SSecurityNo	Employee No	First Name	Last Name	DateOfBirth	Date Employed
AF-23432334	1	Manny	Tomanny	12 Apr 1966	01 May 1999
DQ-65444444	2	Rosanne	Kolumns	21 Mar 1977	01 Jan 2000
GF-54354543	3	Cas	Kade	01 May 1977	01 Apr 2002
JK-34333432	4	Norma	Lyzation	03 Apr 1966	01 Apr 2002
VB-48565444	5	Juan	Tomani	12 Apr 1966	01 Apr 2002
FG-23566553	6	Del	Eats	01 May 1967	01 May 2004

Foreign Key

ORDERS

Primary Key ➡

OrderNo	EmployeeNo	CustomerNo	Supplier	Price	Item
1	1	42	Harrison	$235	Desk
2	4	1	Ford	$234	Chair
3	1	68	Harrison	$415	Table
4	2	112	Ford	$350	Lamp
5	3	42	Ford	$234	Chair
6	2	112	Ford	$350	Lamp
7	2	42	Harrison	$235	Desk

- Primary keys indicated by a key icon

ALLEGHENY
COLLEGE

- Primary Keys
  - *In SQLite, a primary key is a single field or combination of fields that uniquely defines a record. A table can have only one primary key.*
  - Each table in SQLite may have at most one PRIMARY KEY.
  - Uniqueness: information cannot be added to a column if it has already been added before.
  - Primary Keys as NULL values. NULLs are considered distinct from all other values, including other NULLs.
  - It is a violation for an INSERT statement to attempt a modification of table content so that two or more rows have identical primary key values.

Primary Keys
File : sandbox/code_guestDB.txt

Consider these two tables

```
DROP TABLE Residence;
CREATE TABLE Residence (
 id INT PRIMARY KEY,
 room VARCHAR,
 name VARCHAR );

DROP TABLE Guest;
CREATE TABLE Guest (
 id INT PRIMARY KEY,
 firstName VARCHAR,
 lastName VARCHAR,
 residence VARCHAR );
```

Insert data: Copy, then paste twice

```
INSERT INTO Guest VALUES (221,"Sherlock","Holmes","221b Baker Street");
INSERT INTO Guest VALUES (747,"Amelia","Earhart","1 AirWay Street");

INSERT INTO Residence VALUES (221, 0221, "Holmes");
INSERT INTO Residence VALUES (747, 0747, "Earhart");
```

- Foreign Keys
  - *A foreign key is a way to enforce referential integrity within your SQLite database. A foreign key means that values in one table must also appear in another table. The referenced table is called the parent table while the table with the foreign key is called the child table.*
  - An enforced relationship between two tables.
  - information cannot be added unless it *behaves* according to the established relationship between two or more tables.

Foreign Keys
File : sandbox/code_agentCarsDB.txt

Keys
Primary Keys
Foreign Keys
Examples and Code

Add the foreign key code

```
/* Enable foreign keys: automatically check FK constraints */
PRAGMA foreign_keys = ON;

DROP TABLE Car;
CREATE TABLE Car (
name VARCHAR PRIMARY KEY,
registration VARCHAR,
capacity INT,
topSpeed INT
);

DROP TABLE Agent;
CREATE TABLE Agent (
id INT PRIMARY KEY,
lastName VARCHAR,
registration VARCHAR,
worksFor VARCHAR,
FOREIGN KEY(registration) REFERENCES car(name));
```

Setting up Foreign Keys
File : sandbox/code_agentCarsDB.txt

The insertion code using foreign keys

```
INSERT INTO Car values ('AstonMartin', 'MI6', 2, 130);
INSERT INTO Agent values (007, 'Bond', 'AstonMartin', 'MI6');
```

There is no fixed relationship for this insertion. :-(

```
INSERT INTO Agent values (0070, 'Bond', 'Buick', 'MI6');
```

Add the foreign key code to include **Buick** relationship

```
INSERT INTO Car values ('Buick', 'MI6', 5, 60);
```

Relationship established and Insert now works. :-)

```
INSERT INTO Agent values (0070, 'Bond', 'Buick', 'MI6');
```

- The foreign keys establish relationships between tables.

ALLEGHENY
COLLEGE

Keys

Primary Keys

Foreign Keys

Examples and
Code

Adding Data
File : sandbox/code_agentCarsDB.txt

- Add some data into both tables at the same time.

More data to add...

```
INSERT INTO Agent values (008, 'Bennet', 'AstonMartin', 'MI6');
INSERT INTO Agent values (009, 'McBain', 'AstonMartin', 'MI6');
```

- Correct order: First establish the Car table data and then insert the Agent data

Add new foreign key relationship

```
INSERT INTO Car values ('RustyTruck', 'MI7', 10, 30);
INSERT INTO Agent values (10, 'Rufus', 'RustyTruck', 'MI7');
```

Forget-Me-Nots!

How is the database set up?

- .tables (The tables are of the DB)
- .schema (How the data is stored in the tables)

What data is stored in each table?

- SELECT * FROM Agent;
- SELECT * FROM Car;
- (note the '*' for the column wildcard)

just

another

example

Choose a DB to work with queries and foreign keys.

- File: `sandbox/code_carColourDB.txt`:
  Copy and paste lines of code from file to follow examples
- CampusDB from this week already has foreign key code added.

Consider this...
Please see the *sandbox* file for helpful code

- Can you create a similar base where a foreign key has been established?
- Can you write a query to show how the foreign key works?