Introduction to Databases

Author: Ernest Edinboro

Coauthor: "Aneesh Sreedhara"

Date: Summer 2018

Question	Answers
What type of database type does Shodor use?	**Relational**
What DBMS does Shodor use?	**Sequel Pro**

MySQL Building My First Table

SQL	MySQL Syntax	Results
Building Table and Table Info	**Write proper SQL syntax to query the People table using Sequel Pro.**	**What Was the Result**

CREATE TABLE
Step1:
	`CREATE TABLE People (`
	`  id int(11) unsigned NOT NULL auto_increment,`
	`  firstName varchar(255) default NULL,`
	`  lastName varchar(255) default NULL,`
	
	`  favColor varchar(255) default NULL,`
	`  shoeSize decimal (3,1)default NULL,`
	`  PRIMARY KEY  (id)`
	`) ENGINE=InnoDB DEFAULT CHARSET=utf8;`
		Created table called people with specified columns. Empty.

INSERT INTO
Step2:
	`INSERT INTO People (firstName, lastName,favColor, shoeSize)`
	`VALUES`
	`('Ernie','Edinboro','Orange',11.5),`
	`('Gustavo','Nunez-Vivanco','Blue',9.5),`
	`('Chris','Parker', 'Red', 10),`
	`('Huny','Chung','Light Green', 9),`
	`('Aaron','Weeden', 'Purple', 10),`
	`('No','Hope', 'Black', 4),`
	`('Shawn','Kang', 'Red', 10),`
	`('Divya','Aikat','Purple',9.5),`
	`('Michael','Allen Dacany', 'Blue', 12),`
	`('Naylea ','Jacobo', 'Blue', 7.5),`
	`('Mihir','Khadri', 'Red', 10.5),`
	`('Krista', 'Katzenmeyer', 'Cupcake', 8.5),`
	`('Erica','Aiello', 'Yellow', 3),`
	`('Joel','Coldren', 'Green', 13.5),`
	`('Rohin','Shahi', 'Red', 12.5),`
	`('Ameya','Rao', 'Blue', 9.5),`
	`('Luka','Ashe-Jones', 'Orange', 8),`
	`('Ernie','Edinboro', 'Blue', 12),`
	`('Ernie','Edinboro', 'Green', 11),`
	`('Leanne','Chan', 'Green',6),`
	`('Erica','Aiello', 'Purple', 3.5);`
		Inserted values into the above table.

ALTER Table **Step3:**	`ALTER TABLE People` `ADD coinFlip VARCHAR(5) NOT NULL;`	**Adds a column called coinflip in people. Empty.**

Add Data to the newly created coinFlip field.
Step4:
	`INSERT INTO People (id, coinFlip)`
	`VALUES`
	`('1','Tails'),('2','Tails'),('3','Tails'),`
	`('4','Heads'),('5','Heads'),('6','Tails'),`
	`('7','Heads'),('8','Tails'),('9','Heads'),`
	`('10','Heads'),('11','Tails'),('12','Heads'),`
	`('13','Heads'),('14','Heads'),('15','Tails'),`
	`('16','Heads'),('17','Heads'),('18','Heads'),`
	`('19','Heads'),('20','Tails'),('21','Heads')`
	`ON DUPLICATE KEY UPDATE`
	`id=VALUES(id), coinFlip=VALUES(coinFlip);`
		Puts values into coinflip column. Updates duplicates.

MySQL Syntax Guide:
Day 1

SQL	MySQL Syntax	Results
Other usefully SQL for Building Table and Table Info	Write proper SQL syntax to query the People table using Sequel Pro.	What Was the Result
SHOW TABLE	`SHOW TABLES;`	Shows the table names.
DESCRIBE TABLE	`DESCRIBE People;`	Shows fields, types, null, default, and extras for the table.
DROP TABLE Don't run unless you really want to get rid of a table and all its data. You will have to start over. DO NOT RUN QUERY!	`DROP People;`	DO NOT RUN this query unless you want to destroy a table and data. Once done it cannot be undone.

NOT NULL	`Example statement:`
	`...`
	`id int(11) unsigned NOT NULL auto_increment,`
	`...`
		Value cannot be null

PRIMARY KEY	`Example statement:`
	`...`
	`PRIMARY KEY (ID)`
	`...`
		The ID column is now the unique id.

AUTO_INCREMENT	Example statement: ... id int(11) unsigned NOT NULL auto_increment, ...	**Unique ID automatically goes up as more values are added.**
DELETE **Be careful! DO NOT RUN QUERY!**	DELETE People;	**Do not run query**
Useful SQL Syntax		
UPDATE	UPDATE People set lastName='Happiness' WHERE id=6;	**Updates lastName on id 6 to Happiness.**
SELECT	SELECT firstName, lastName FROM People;	**Selects data from columns firstName and lastName and stores them in result data set.**
WHERE	SELECT lastName FROM People WHERE favColor= 'Red';	**Takes only last name data whose favorite color is red.**
ORDER BY	SELECT id, lastName FROM People WHERE favColor= 'Red' ORDER BY id;	**Orders the result set by id.**
AND, OR, Not	SELECT id, lastName FROM People WHERE favColor= 'Red' AND id > 4; SELECT id, lastName FROM People	**Results in data that meets all specifications.**

	`WHERE favColor= 'Red' OR id > 4;`
	
	`SELECT id, lastName`
	`FROM People`
	`WHERE NOT id > 4;`	

Count, Avg, Sum	`SELECT COUNT(id)`
	`FROM People`
	`WHERE favColor= 'Red' OR id > 4;`
	
	`SELECT AVG(id)`
	`FROM People`
	`WHERE favColor= 'Red' OR id > 4;`
	
	`SELECT SUM(id)`
	`FROM People`
	`WHERE favColor= 'Red' OR id > 4;`
		Count returns the number of rows that meet the specifications
		
		AVG returns the average of the values of the rows that meet the criteria.
		
		SUm returns the sum of the rows that meet the specifications.

LIKE	`SELECT firstName, lastName`
	`FROM People`
	`WHERE favColor LIKE 'R%';`
		Returns data that meets specifications anf favColor beng anything that starts with R.

IN	`SELECT firstName, lastName`
	`FROM People`
	`WHERE favColor IN ('Red', 'Green');`
		Gets people that like red, and people that like green.

BETWEEN	`SELECT firstName, lastName`
	`FROM People`
	`WHERE id BETWEEN 1 AND 5;`
		Gets peopl with id between 1 and 5

ALIAS	`SELECT firstName`
	`AS name1`
	`FROM People;`
		So basically, you can call it with both names now.

SQL Operators	**Best to use Arithmetic and Comparison Operators on shoeSize field**	
Arithmetic Operators		

Add	`UPDATE People`
	`SET shoeSize=30 + 20`
	`WHERE id=6;`
		`50`

Subtraction	`UPDATE People`
	`SET shoeSize=30 - 20`
	`WHERE id=6;`
		`10`

Division	`UPDATE People`
	`SET shoeSize=30 / 20`
	`WHERE id=6;`
		`30/20`

Multiplication	`UPDATE People`
	`SET shoeSize=30 * 20`
	`WHERE id=6;`
		`600`

Modulos	`UPDATE People`
	`SET shoeSize=30 % 20`
	`WHERE id=6;`
		`10`

Comparison Operators		

Equal to	`SELECT firstName, lastName`
	`FROM People;`
	`where id=6;`
		`Selects people with id = 6`

Greater than >	`SELECT firstName, lastName`
	`FROM People;`
	`where id>6;`
		`Selects people with id>6`

Less than <	`SELECT firstName, lastName`
	`FROM People;`
	`where id<6;`
		`Selects people with id < 6`

Greater than equal to >=	`SELECT firstName, lastName`
	`FROM People;`
	`where id>=6;`
		`Selects people with id >= 6`

Less than equal to <=	`SELECT firstName, lastName`
	`FROM People;`
	`where id<=6;`
		`Selects people with id <= 6`

Not equal to <>	`SELECT firstName, lastName`
	`FROM People;`
	`where id<>6;`
		`Selects people with id <> 6`

SQL Data Types

List 10 SQL Data Types	What types of Data do they Store?	
Binary	`Binary string`	
Boolean	`True or False`	
REAL	`Approximate numerical`	
Date	`Year month and day variables`	
XML	`XML data`	
Array	`Collection of elements`	

Writing SQL queries

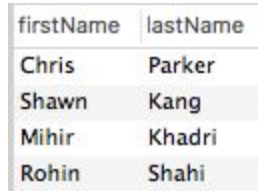
SQL MINI Challenge **Cannot use INT or VARCHAR**	**Write SQL query to answer the questions**	**Result**

What is Erica's favorite color?	`SELECT favColor`
	`FROM People`
	`WHERE firstName = 'Erica';`
		`Yellow`
		`Purple`

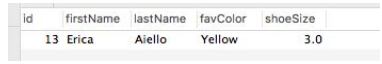
Whose shoe size is 3?
Show First and last name
	`SELECT firstName, lastName`
	`FROM People`
	`WHERE shoeSize = 3;`
		`Erica Aiello`

Whose favorite color is Cupcake?
Show First Name only
	`SELECT firstName`
	`FROM People`
	`WHERE favColor = 'Cupcake';`
		`Krista`

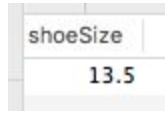
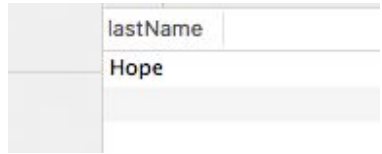
What is the average shoe size?	`Select AVG(shoeSize)` `FROM People;`	`9.09524`

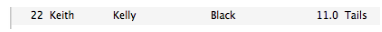
Whose favorite color is Red?
Show Last Name only ORDER BY ASCending order.
	`Select firstName, lastName`
	`FROM People`
	`WHERE favColor = 'Red';`
		

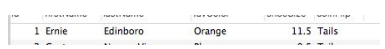
What is the favorite color of people with shoe size 10?	`Select favColor`
	`FROM People`
	`WHERE shoeSize = 10;`
		

Select all data for user(s) id =13	`Select *`
	`FROM People`
	`WHERE id = 13;`
		

Select first name for user(s) where favorite color is red	`select firstName`
	`FROM People`
	`WHERE favColor= 'Red';`
		

Select favorite color where shoe size greater than or equal to 12	`select favColor from people where shoeSize >= 12;`	
Select shoe size where first name = Joel	`select shoeSize from People where firstName = 'Joel';`	
Select lastname where favorite color is black	`select lastName from People where favColor = 'Black';`	

Insert into People Keith,Kelly, Black, 11, Tails	`INSERT into People (firstName, lastName, favColor, shoeSize, coinFlip)`
	`VALUES`
	`('Keith', 'Kelly', 'Black', 11, 'Tails');`
		

Update First name of id 1 to Ernest	`update People`
	`set firstName = 'Ernie'`
	`where id=1;`
		

MySQL Syntax Guide: <- This is the same as below so I just did it below.
Day 2

Database Interaction	MySQL Syntax using People table	What where the results?

PRIMARY KEY	`Example statement:`
	`...`
	`PRIMARY KEY (ID)`
	`...`
		The ID column is now the unique id.

FOREIGN KEY		
JOIN		
INNER JOIN		
LEFT JOIN		
RIGHT JOIN		
GROUP BY		

SQL Snippet for creating Relational DB	`CONSTRAINT CoinFlip_ibfk_1`
	`FOREIGN KEY (peopleId) REFERENCES People (id)`
	`ON DELETE CASCADE ON UPDATE CASCADE,`	

SQL MINI Challenge	SQL Syntax used	Result
?		
?		
?		
?		

MySQL Syntax Guide:
Day 2

Database Interaction	MySQL Syntax using newly created relational database People, coinFlips, color, table, shoes	What where the results?

PRIMARY KEY	`...`
	`PRIMARY KEY (id),`
	`...`
		The id row is designated as the unique row.

FOREIGN KEY	`...`
	`FOREIGN KEY (peopleId) REFERENCES People(id)`
	`...`
		The peopleId column now corresponds to the id column from the people table.

JOIN	`SELECT People.id, People.firstName, People.lastName, CoinFlip.result`
	`FROM People`
	`JOIN CoinFlip ON CoinFlip.peopleId=People.id;`
		All records are added, with mathing ones merged.

INNER JOIN	`SELECT People.id, People.firstName, People.lastName, CoinFlip.result`	Only merged matching records are displayed.

	FROM People INNER JOIN CoinFlip ON CoinFlip.peopleId=People.id;	
LEFT JOIN	SELECT People.id, People.firstName, People.lastName, CoinFlip.result FROM People LEFT JOIN CoinFlip ON CoinFlip.peopleId=People.id;	All records from the left and matching records from the right are displayed, with matching ones getting merged.
RIGHT JOIN	SELECT People.id, People.firstName, People.lastName, CoinFlip.result FROM People RIGHT JOIN CoinFlip ON CoinFlip.peopleId=People.id;	All records from the right and matching records from the Left are displayed, with matching ones getting merged.
GROUP BY	SELECT COUNT(id), color FROM Color GROUP BY color;	Groups the result set into color groups.
SQL Snippet for creating Relational DB	CONSTRAINT CoinFlip_ibfk_1 FOREIGN KEY (peopleId) REFERENCES People (id) ON DELETE CASCADE ON UPDATE CASCADE,	peopleId is now corresponding with id from the People dataset. A delete or update of a row changes both tables.
