Introduction to Databases

Author: Hudson

Coauthor: Ernest Edinboro

Date: Summer 2020

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

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;`
		A new table was created, 0 rows were affected

INSERT INTO
Step2:
	`INSERT INTO People (fName, lName,favColor, shoeSize)`
	`VALUES`
	`('Ernie','Edinboro','Orange',11.5),`
	`('Aras','Rose','Blue',9.5),`
	`('Inaayah','Galloway', 'Red', 10),`
	`('Ruby','Chung','Light Green', 9),`
	`('Amit','Lloyd', 'Purple', 10),`
	`('Jerry','Ford', 'Black', 4),`
	`('Miruna','Woods', 'Red', 10),`
	`('Jamie','Pemberton','Purple',9.5),`
	`('Avani','Lister', 'Blue', 12),`
	`('Hanifa ','Phelps', 'Blue', 7.5),`
	`('Madina','Lowe', 'Red', 10.5),`
	`('Elsie', 'Wise', 'Cupcake', 8.5),`
	`('Jacques','Oakley', 'Yellow', 3),`
	`('Andrea','Waller', 'Green', 13.5),`
	`('Hadi','Washington', 'Red', 12.5),`
	`('Keiron','Schofield', 'Blue', 9.5),`
	`('Farrell','Mosley', 'Orange', 8),`
	`('Markus','Collins', 'Blue', 12),`
	`('Ernie','Edinboro', 'Green', 11),`
	`('Ernie','Edinboro', 'Blue',6),`
	`('Fynley','Merce', 'Purple', 3.5);`
		We created 4 rows, and populated them with 21 ids

ALTER Table **Step3:**	`ALTER TABLE People` `ADD coinFlip VARCHAR(5) NOT NULL;`	**We added another column to the table**

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);`
		Skip

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 TABLES	`SHOW TABLES`	The tables are displayed on the top of the screen
DESCRIBE TABLE	`DESC People;`	Displays the head for tch collum, theur var type, and other parameters
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 Table people;`	DO NOT RUN this query unless you want to destroy a table and data. Once done it cannot be undone.

NOT NULL	`ALTER TABLE People`
	
	`MODIFY firstName varchar(255) NOT NULL;`
		Makes first name have to have a value

PRIMARY KEY	`PRIMARY KEY (columnName)`	Makes the column in the parentheses a primary key. A primary key must contain a unique value

AUTO_INCREMENT	`ColumnName int AOTO_INCREMENT`	**Counts up, creating a unique int every time**
DELETE **Be careful! DO NOT RUN QUERY!**	`DELETE FROM People WHERE firstName = “ernie”`	**Do not run query**
Useful SQL Syntax		

UPDATE	`UPDATE People`
	`SET favColor = “green”`
	`WHERE id =12;`
		Changes the row with the id 12 favColor to green

SELECT	`SELECT lastName FROM People;`	**Displays the column lastName**

WHERE	`UPDATE People`
	`SET favColor = “green”`
	`WHERE id =12;`
		Where helps filter conditions

ORDER BY	`SELECT coinFlip`
	`FROM People`
	`ORDER BY id ASC;`
		Displays coinFliop column in ascending order

AND, OR, Not	`SELECT coinFlip`
	
	`FROM People`
	
	`WHERE firstName='Ernie' AND lastName ='Edinboro'`
		Displays the coin flip results for ernie

LIKE	`SELECT * FROM People`
	
	`WHERE favColor LIKE '____'`
		Selects all 4 letter colors

IN	`SELECT * FROM People`
	
	`WHERE favColor IN ('Blue','Red','Green')`
		Selects people who's favorite color is blue, red, and green

BETWEEN	`SELECT *`
	
	`FROM People`
	
	`WHERE id BETWEEN 10 and 20;`
		Selects the rows that have an id between 10 and 20

ALIAS	`SELECT firstName AS "first name"`
	
	`FROM People;`
		Alias temporarily changes the name of anything to make displaying your information more clear

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

Add	`SELECT firstName`
	
	`from People`
	
	`WHERE id = 7+6`
		`First name with the id of 13 (7+6) is displayed`

Subtraction	`SELECT firstName`
	`from People`
	`WHERE id = 7-6`
		First name with the id of 1 (7-6) is displayed

Multiplication	`SELECT firstName`
	`from People`
	`WHERE id = 3 * 5`
		First name with the id of 15 (3*5) is displayed

Division	`SELECT firstName`
	`from People`
	`WHERE id = 30 /3`
		First name with the id of 10 (30/3) is displayed

Modulos	`SELECT firstName`
	`from People`
	`WHERE id = 9 % 5`
		First name with the id of 4 (9%5) is displayed

Comparison Operators		
Equal to	`SELECT *` `from People`	Selects all from people where the id is equal to 4

	`WHERE id = 4`	

Greater than >	`SELECT *`
	`from People`
	`WHERE id > 4`
		Selects all from people where the id is greater than 4

Less than <	`SELECT *`
	`from People`
	`WHERE id < 4`
		Selects all from people where the id is less than than 4

Greater than equal to >=	`SELECT *`
	`from People`
	`WHERE id >= 4`
		Selects all from people where the id is greater than or equal to 4

Less than equal to <=	`SELECT *`
	`from People`
	`WHERE id <= 4`
		Selects all from people where the id is less than than or equal 4

Not equal to <>	`SELECT *`
	`from People`
	`WHERE id <> 4`
		Selects all from people except 4

Useful Math Functions		
Count	`SELECT COUNT(id)` `FROM People`	Returns the number of id column there are (21)
Avg	`SELECT AVG (shoeSize)` `FROM People`	Calculates the average shoe size 9.05
Sum	`SELECT SUM(shoeSize)` `FROM People`	Calculates the sum of all the shoe sizes (191)

SQL Data Types

List 10 SQL Data Types You cannot user INT or VARCHAR	What types of Data do they Store?	
CHAR()	A string with a fixed length, can be between 0 and 255, default 1	`ALTER TABLE People` `MODIFY FirstName char(5) NOT NULL;`
TINYBLOB	Used to store binary large objects, can be up to 255	`ALTER TABLE People` `MODIFY lastNameTINYBLOB NOT NULL;`
TEXT()	Stores a string up to 65,535	`ALTER TABLE People`

		MODIFY famlyBiography Text (65535)NOT NULL;
BIT()	**Can hold a number, size 1 to 64**	ALTER TABLE People MODIFY shoeSize BIT(4)
INT()	**An integer that can hold up to a signs range of -1247483648 to 1247483647 or a unsigned range of 0 to 4294967295**	peopleID int(11) unsigned NOT NULL,
FLOAT()	**Stores a floating point number if number in parentheses**	shoeSize DEC(3)default NULL,
DEC()	**Stores a decimal, the first number is the total number of numbers, the second is the number after the parentheses**	shoeSize DEC(3,1)default NULL,
DATE	**Stores info in the format is YYYY-MM-DD**	visitDate DATE default NULL
BOOL	**Stores boolean value, 0 if false, 1 if true**	checkBox BOOL NOT NULL
SMALLINT	**As in small int, is like int but smaller. Signed range -32768 to 32767 and unsigned range 0 to 16777215**	userAge SMALLINT(3)

Writing SQL queries

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

What is Elsie's favorite color?	`SELECT favColor`
	`FROM People`
	`WHERE firstName = 'Elsie'`
		favColor is displayed with green under it

Whose shoe size is 3? **Show First and last name**	`SELECT firstName, lastName`	FirstName, then lastName were displayed

	FROM People where shoeSize = 3;	with Jacques Oakley under it
Whose favorite color is Cupcake? Show First Name only	SELECT firstName FROM People where favColor = Cupcake	Favorite color was displayed with Elise under it
What is the average shoe size?	SELECT AVG (shoeSize) FROM People	average shoe size 9.05
Whose favorite color is Red? Show Last Name only ORDER BY ASCending order.	SELECT lastName from People WHERE favColor = 'Red' ORDER BY id ASC;	Last name was displayed with Galloway, Woods,Lowe, and Washington immediately under it
What is the favorite color of people with shoe size 10?	SELECT favColor from People WHERE shoeSize = 10	favColor was displayed with Red, Purple, and red displayed immediately under it
Select all data for user(s) id =13	SELECT *	A whole row including id first name, last name, etc. was

	`from People`
	
	`WHERE id = 13`
		displayed with the name jacques Okely, Yellow, etc. displayed

Select first name for user(s) where favorite color is red	`SELECT firstName`
	
	`from People`
	
	`WHERE favColor = 'Red'`
		firstName with the names Innaayah Miruna Madina and Hadi is displayed

Select favorite color where shoe size greater than or equal to 12	`SELECT favColor`
	
	`from People`
	
	`WHERE shoeSize >=12`
		Blue, green, red, blue

Select shoe size where first name = Farrell	`SELECT shoeSize`
	
	`from People`
	
	`WHERE firstName = 'Farrell'`
		Shoe Size = 8

Select lastname where favorite color is black	`SELECT lastName`
	
	`from People`
	
	`WHERE favColor = 'Black'`
		Last name: Ford

Insert into People Jimmy, Johns, Black, 11, Tails	`INSERT INTO People (firstName, lastName, favColor, shoeSize, coinFlip)`
	
	`VALUE`
		Adds all that to the People table

	`('Jimmy', 'Johns', 'Black', 11, 'Tails');`	

Update First name of id 1 to Ernest	`UPDATE People`
	
	`SET firstName = 'Ernie'`
	
	`WHERE id = 1;`
		`Nothing happened cuz id 1 was already ernie`

STOP, you are done with Day 1

MySQL Syntax Guide:
Day 2 or Day 3

Database Interaction	MySQL Syntax using People table	What where the results?
PRIMARY KEY	`PRIMARY KEY (id)`	`Makes is the primary key`
FOREIGN KEY	**FOREIGN KEY (id) REFERENCES People(PeopleID)**	`Links id and PeopleID`
JOIN	`INNER JOIN People ON people.ID=coinFlip.ID`	`Joins the similarities`

		between two tables
INNER JOIN	`This acts the same as join and uses the same syntax`	
LEFT JOIN	`LEFT JOIN People` `ON People.fName=coinFlip.fname;`	Joins all matching elements to people
RIGHT JOIN	`RIGHT JOIN coinFlip` `ON People.lName=coinFlip.lName`	Joins all matching elements to coinFlip
GROUP BY	`GROUP BY fName`	Collects duplicate rows

SQL Snippet for creating Relational DB	`CONSTRAINT CoinFlip_ibfk_1`
	`FOREIGN KEY (peopleId) REFERENCES People (id)`
	`ON DELETE CASCADE ON UPDATE CASCADE,`
		This piece of code ties peopleId to id, amd sets wuo guidelines for updating the code

SQL MINI Challenge	**SQL Syntax used**	**Result**
? Guess you don't have to do anything		
? Guess you don't have to do anything		
? Guess you don't have to do anything		
? Guess you don't have to do anything		
? Guess you don't have to do anything		
? Guess you don't have to do anything		