Informatics Institute Of Technology

Database Systems

5COSC020C.1

Part A

Name - Nendunmullage Tharusha Rukshan Ravindra

Uow Number – W1830179

IIT Number – 20200459

Tutorial Group – K

Table of Contents

1) CONCEPTUAL ERD 3

Assumptions 4

2) Data Dictionary to document how I identified the entities 5

3) A data dictionary to document how I identified the relationships and multiplicities 7

4) Data dictionary to document how I identified the attributes and primary keys 12

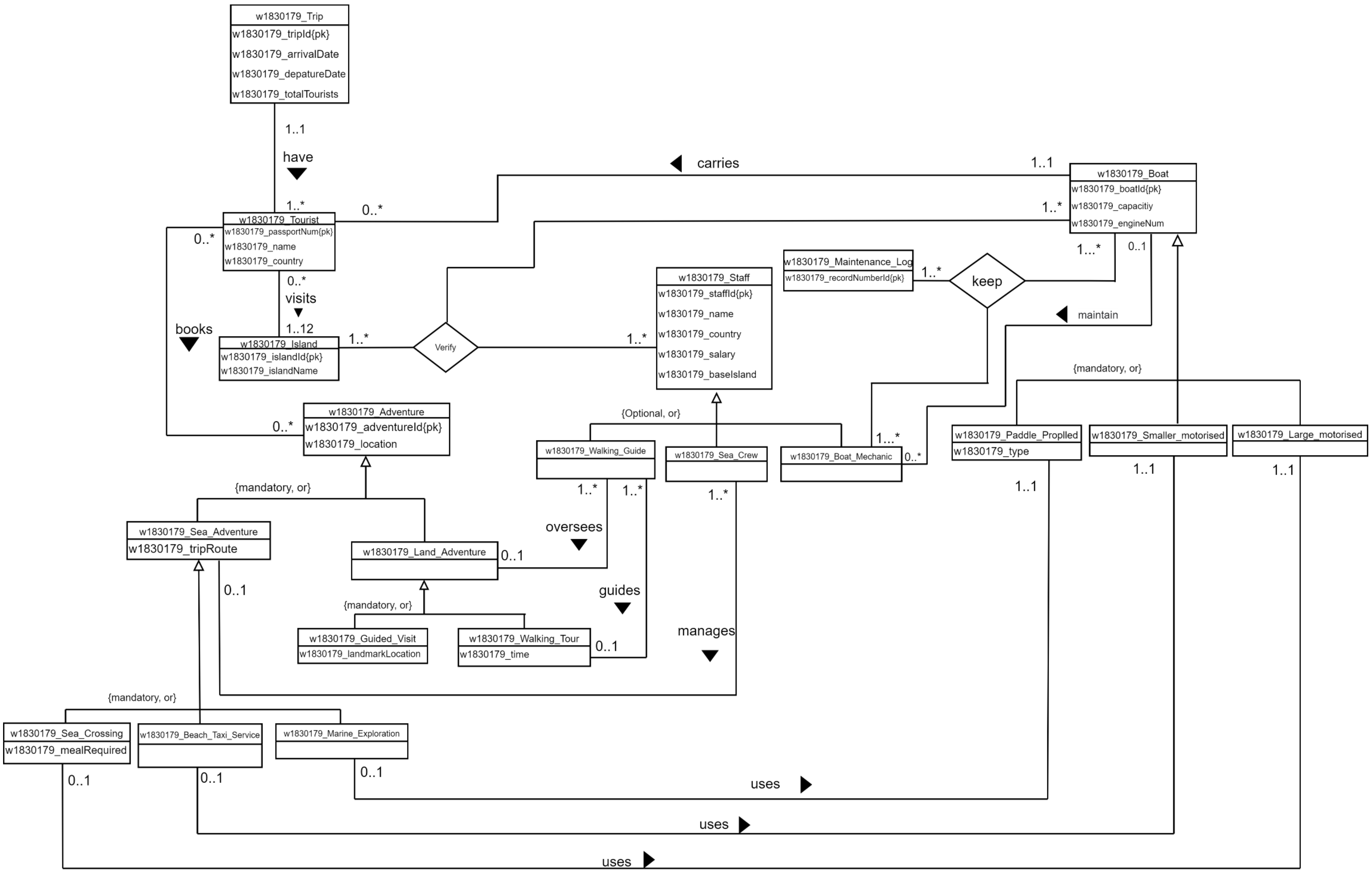
5) The logical ERD 14

6) SQL query 15

7) Comparative analysis table Between Relational Databases vs. NoSQL databases with a view to inform the decision-making of the management of a firm. 23

7.1) Citations 26

1) CONCEPTUAL ERD



Assumptions

1. I assume that one trip can be arranged for multiple tourists. As an example, tourists can come in as groups such that causes Archipelago Crazy only keep one record of that entire tour group.

2. I assume that there can be more staff types except for the three types given in the case study. So, I assume that participant constraint should be mandatory. And the disjoint constraint should be or because it strictly says that roles are specialized one staff can only be one, they cannot overlap.

3. I assume that they maintenance a maintain log because in the case study they have given they should record every service carefully and I assume that it should done by boat mechanics.

4. I assume that guided visit does not requires a walking guide (in the case study they have not specifically mentioned it. But have specifically mentioned a walking tour has a guide) guided visit can be a guideline tour which gives only guidelines before the tour.

5. I assume that every tourist must at least visit one island.

2) Data Dictionary to document how I identified the entities

Entity Name	**Brief Description**
w1830179_Tourist	General term describing all tourists who arrange trips from ArchipelagoCrazy.
w1830179_Boat	General term describing all boats which ArchipelagoCrazy use to provide services that tourist require.
w1830179_Adventure	General term describing all adventures ArchipelagoCrazy provides.
w1830179_Staff	General term describing all staff employed by ArchipelagoCrazy.
w1830179_Island	General term describing all islands which ArchipelagoCrazy organizes trips.
w1830179_Trip	General term describing all trips which ArchipelagoCrazy organizes.
w1830179_Maintenances_Log	General term describing all maintenance records which ArchipelagoCrazy keep on their boats.

General entity	**Specialized entity**	**Brief Explanation**
w1830179_Boat	w1830179_Large_Motorised	A general term to describe boats that can carry higher number of passengers in it. This boat type used to carry passengers from the mainland to the islands.
	w1830179_Small_Motorised	A general term to describe boats that can carry between only 6 to 8 people in it. And this is used to provides a local taxi service.
	w1830179_Paddle_propelled	A general term to describe boats that is used carry passengers to more remote bays and creeks.
w1830179_Adventure	w1830179_Sea_Adventure	A general term to describe all adventures which take place at sea. This entity has three more specialized entities.
	w1830179_Land_Adventure	A general term to describe all adventures which take place at land.
w1830179_Sea_ Adventure	w1830179_Sea_Crossing	Traveling between islands using lager motorised boat, with having a meal on board if required.
	w1830179_Beach_Taxi Service	A general term to describe travelling among beaches using smaller motorised boats.
	w1830179_Marine_Exploration	A general term to describe exploring remote areas in sea using a paddle-propelled boat.
w1830179_Land_Adventure	w1830179_Guided_Visit	A general term to describe visiting landmarks in the area.
	w1830179_Walking_Tour	A general term to describe land tours which can be nocturnal if required and this requires guide to guide them
w1830179_Staff	w1830179_Sea_Crew	A general term to describe Staff which organizes and manages sea adventures for ArchipelagoCrazy.
	w1830179_Walking_Guide	A general term to describe Staff which employs by ArchipelagoCrazy to organizes and manages land adventures for.
	w1830179_Boat_Mechanic	A general term to describe Staff which maintenance all the boats which used by ArchipelagoCrazy.

3) A data dictionary to document how I identified the relationships and multiplicities

Entity name	Multiplicity	Relationship	Multiplicity	Entity name	Brief justifications for the multiplicity (4 statements for each relationship)
w1830179_Trip	1..1	have	1.. *	w1830179_Tourist	One trip may have one tourist on it. without a single tourist a trip may not exist.
					One trip may have many tourists on it. Because tourist can come in as groups.
					One tourist must have one trip.
					One tourist can only have up to one trip.
W1830179_Tourist	0.. *	visits	1..12	W1830179_Island	One tourist visits at least one island. Tourist must at least visit one island.
					One tourist can only visit up to twelve islands.
					One island may not have any tourist visiting it.
					One island may have many tourists visiting it.
W1830179_Tourist	0.. *	books	0.. *	W1830179_adventure	One tourist can book no adventure.
					One tourist can book many adventures.
					One adventure can be book by no tourist.
					One adventure can be book by many tourists.

W1830179_Boat	1..1	carries	0.. *	W1830179_Tourist	One boat may be not carrying any tourist on it. Boats can go without tourist to get serviced.
					One boat can carry many tourists on it.
					One tourist can be carried by one boat.
					One tourist can be carried by only one boat.
w1830179_Walking_Guide	1.. *	oversee	0..1	w1830179_Land_Adventure	One walking guide may not oversee any land adventure.
					One walking guide can only oversee up to one land adventure.
					One land adventure can be overseen by at least one walking guide.
					One land adventure can be overseen by many walking guides.
w1830179_Walking_Guide	1.. *	guides	0..1	w1830179_Walking_Tour	One walking guide may not guide any walking tour.
					One walking guide can only guide up to walking tour.
					One walking tour can be guided by at least one walking guide.
					One walking tour can be guided by many walking guides.
W1830179_Sea_Crew	1.. *	manages	0..1	W1830179_Sea_Adventure	One sea crew may not manage any sea adventure.
					One sea crew can only manage up to one sea adventure.
					One sea adventure can be managed by at least one sea crew.
					One sea adventure can be managed by many sea crew.

w1830179_Marine_Exploration	0.. 1	uses	1..1	w1830179_Paddle_propelled	One marine exploration uses at least one paddle propelled boat.
					One marine exploration uses up to one paddle propelled boat.
					One paddle propelled boat may not be used in any marine exploration adventure.
					One paddle propelled boat can be used in one marine exploration adventure.
W1830179_Boat	0.. 1	maintain	1.. *	w1830179_Boat_Mechanic	One boat maintained by at least one boat mechanic.
					One boat can be maintained by many boat mechanics.
					One boat mechanic can maintain no boats.
					One boat mechanic can maintain up to one boat.
w1830179_Beach_Taxi_Service	0.. 1	uses	1..1	w1830179_Smaller_Motorised	One beach taxi service uses at least one smaller motorised boat.
					One beach taxi service uses up to one smaller motorised boat.
					One smaller motorised boat may not be used in any beach taxi service.
					One smaller motorised boat can be used in one beach taxi service.

w1830179_Sea_Crossing	0.. 1	uses	1..1	w1830179_Large_Motorised	One Sea crossing adventure uses at least one Large motorised boat.
					One Sea crossing adventure uses up to one Large motorised boat.
					One Large motorised boat may not be used in any Sea crossing adventure.
					One Large motorised boat can be used in one Sea crossing adventure.
w1830179_Staff	1.. *	Verify	1.. *	w1830179_Boat	One staff member verifies at least one boat in one island.
			1.. *	w1830179_Island	One staff member may verify many boats in many islands
w1830179_Boat	1.. *	Verify	1.. *	w1830179_Staff	One boat at least verifies by one staff in one island.
			1.. *	w1830179_Island	One boat may be verified by many staff members at many islands.
w1830179_Island	1.. *	Verify	1.. *	w1830179_Staff	One island verifies at least one boat from one staff member.
			1.. *	w1830179_Boat	One island may verify many boats from many staff members.
w1830179_Boat_Mechanic	1.. *	Keeps	1.. *	w1830179_Maintenances_Log	One boat mechanic may at least keep one maintenance log for one boat.
			1.. *	w1830179_Boat	One boat mechanic may keep many maintenance logs for many boats.

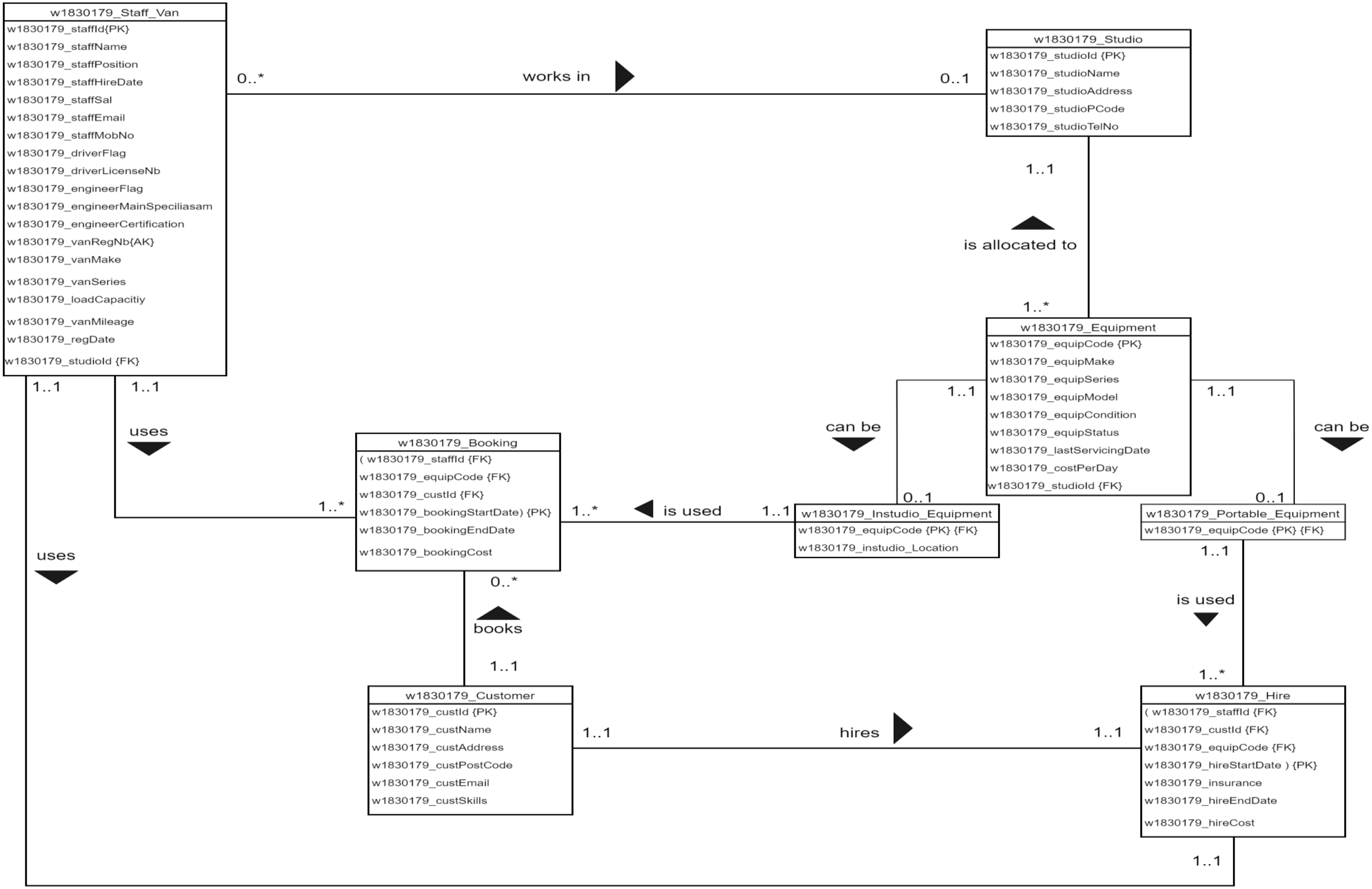
w1830179_Boat	1.. *	Keeps	1.. *	w1830179_Maintenances_Log	One boat at least keeps one maintenances log from one boat mechanic.
			1.. *	w1830179_Boat_Mechanic	One boat may keep many maintenances log from many boat mechanics.
w1830179_Maintenances_Log	1.. *	Keeps	1.. *	w1830179_Boat	One maintenances log at least kept for one boat from one boat mechanic.
			1.. *	w1830179_Boat_Mechanic	One maintenances log can be kept from many boat mechanics for many boats.

4) Data dictionary to document how I identified the attributes and primary keys

Entity name	Attributes for this entity (include PK)	Justification
w1830179_Trip	w1830179_tripId{pk}	Uniquely identifies a trip.
	w1830179_arrivalDate	Arrival date of the tourist or the tourist groups.
	w1830179_depatureDate	Departure date of the tourist or the tourist groups.
	w1830179_totalTourists	Number of tourists involved in the trip.
w1830179_Tourist	w1830179_passportNum{pk}	Uniquely identifies a tourist.
	w1830179_name	Name of the tourist.
	w1830179_country	Country of the tourist.
w1830179_Island	w1830179_islandId{pk}	Uniquely identifies an island.
	w1830179_islandName	Name of the island.
w1830179_Adventure	w1830179_adventureId{pk}	Uniquely identifies an adventure.
	w1830179_location	Adventure held location.
w1830179_Sea_Adventure	w1830179_boatId	Id of the boat used in the sea adventure.
	w1830179_tripRoute	Trip rout of the adventure.
w1830179_Guided_Visit	w1830179_landmarkLocation	Location of the visiting landmark.
w1830179_Walking_Tour	w1830179_time	Time of the tour nocturnal or not nocturnal.
w1830179_Sea_Crossing	w1830179_mealRequired	Meal required in the adventure it can be null as well.
w1830179_Maintenance_Log	w1830179_recordNumberId{pk}	Uniquely identifies a maintenance log.
w1830179_Boat	w1830179_boatId{pk}	Uniquely identifies a boat
	w1830179_capacitiy	Capacity of the boat.
	w1830179_engineNum	Engine number.
w1830179_Paddle_Proplled	w1830179_type	Type of the paddle propelled boat.

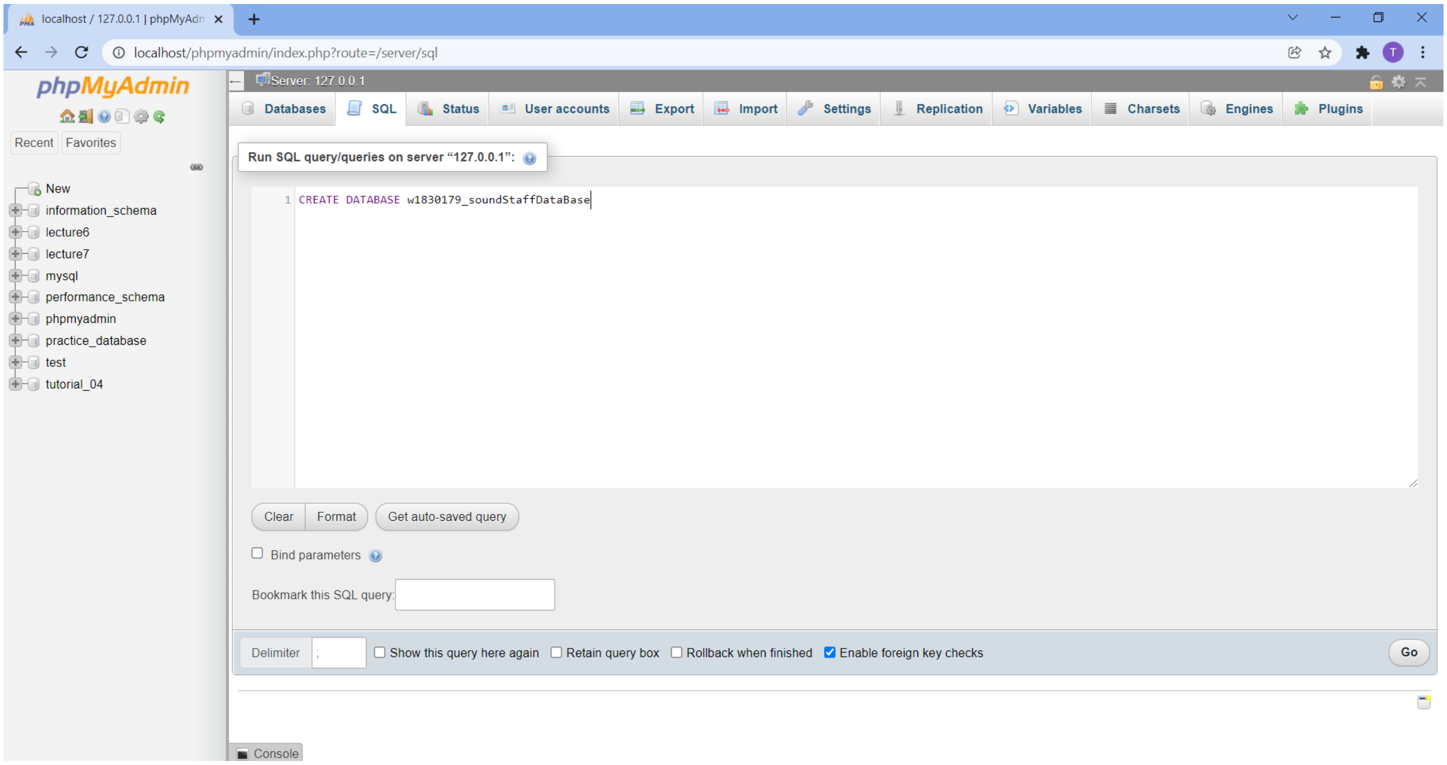
w1830179_Staff	w1830179_staffId{pk}	Uniquely identifies a staff member.
	w1830179_name	Name of the staff member
	w1830179_country	Original country of the staff member
	w1830179_salary	Salary of the staff member

5) The logical ERD



6) SQL query

Creating the database



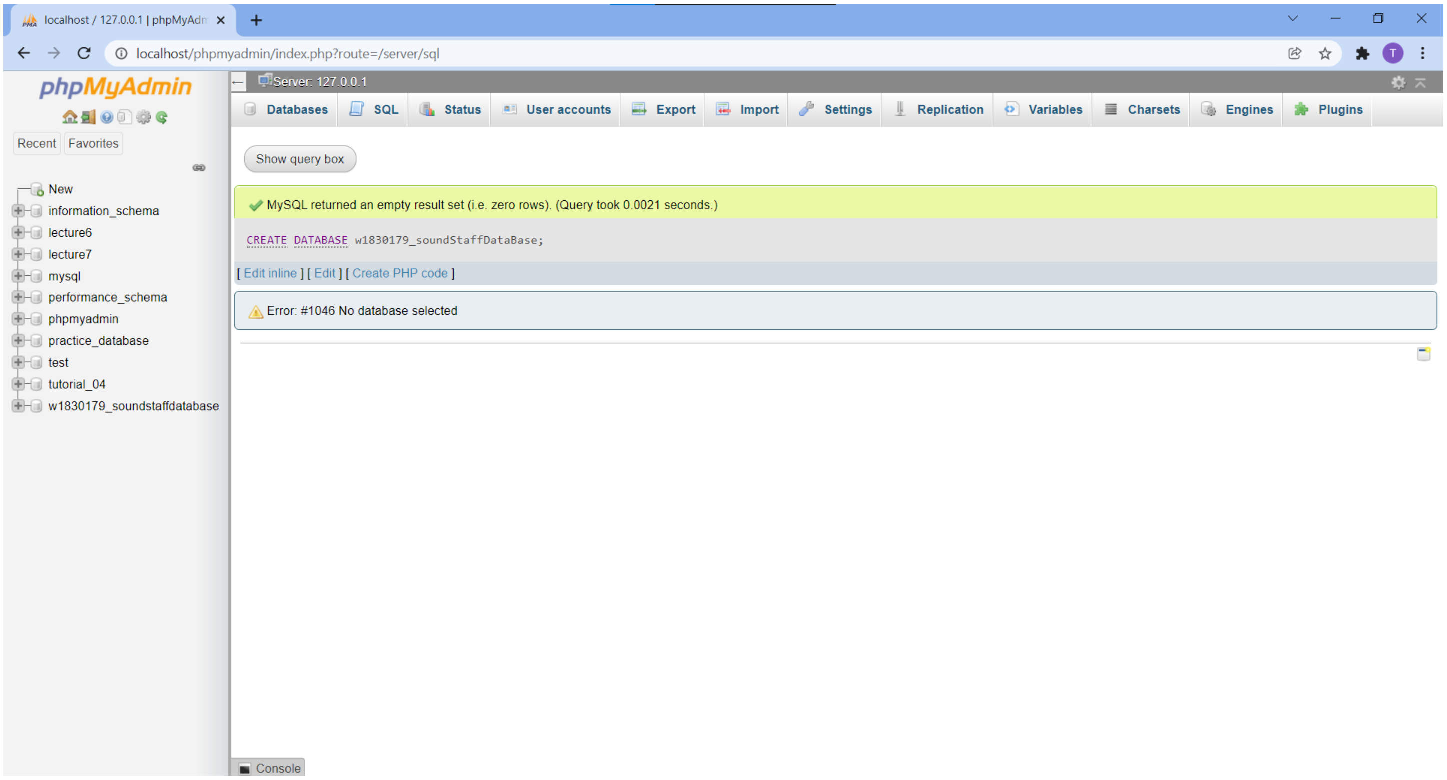
Run SQL query/queries on server "127.0.0.1":

```
CREATE DATABASE w1830179_soundStaffDataBase
```

□ Bind parameters

Bookmark this SQL query:

Delimiter ; □ Show this query here again □ Retain query box □ Rollback when finished ☑ Enable foreign key checks



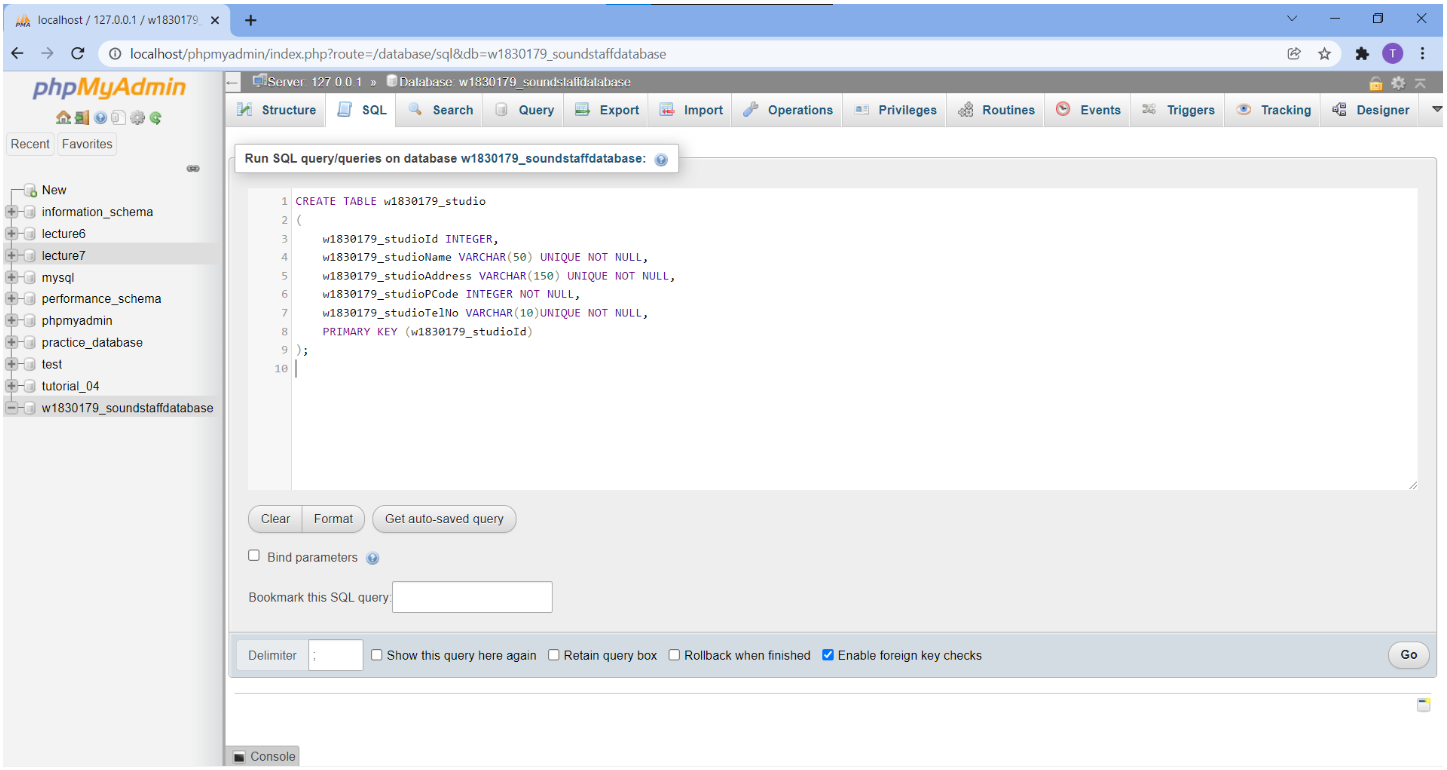
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0021 seconds.)

```
CREATE DATABASE w1830179_soundStaffDataBase;
```

[Edit inline] [Edit] [Create PHP code]

⚠ Error: #1046 No database selected

Creating the studio table



Run SQL query/queries on database w1830179_soundstaffdatabase:

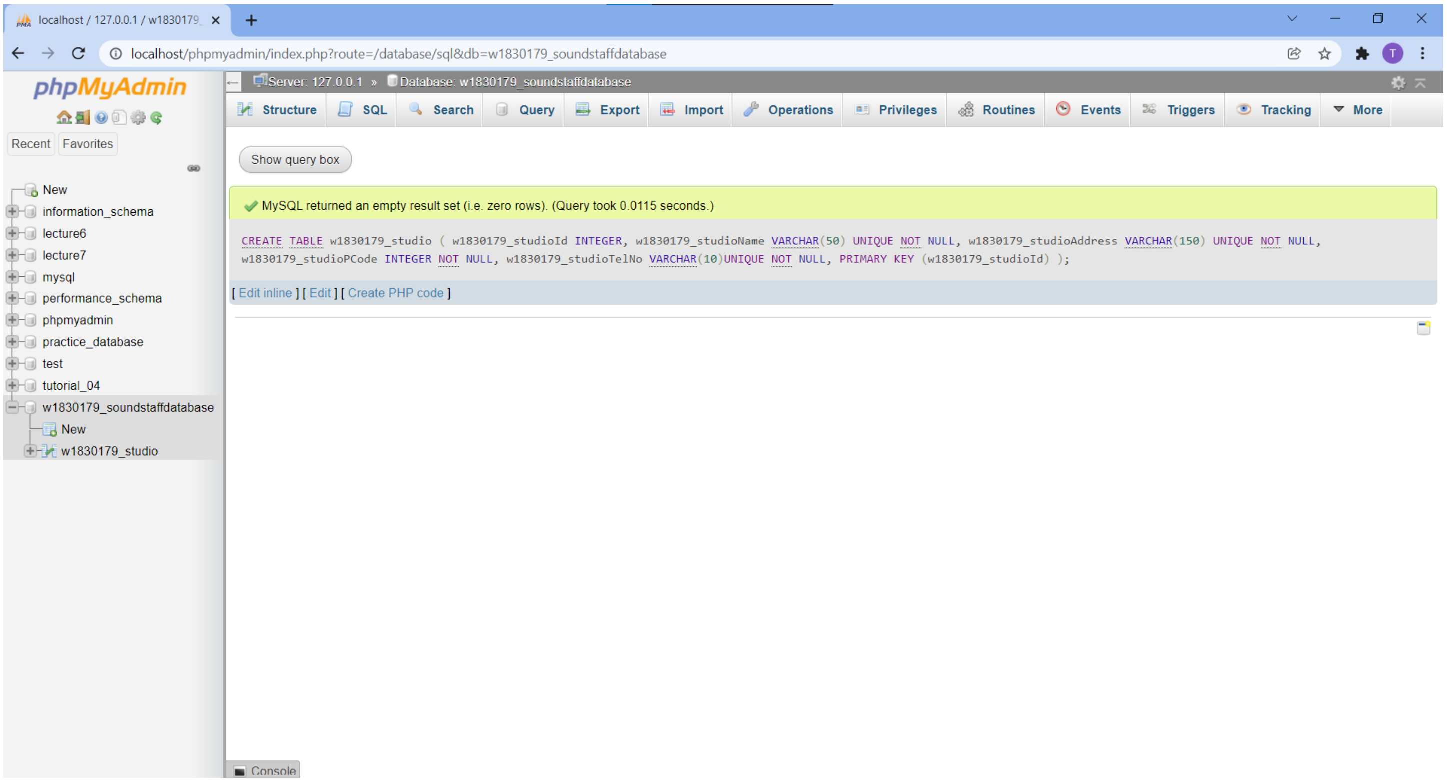
```
CREATE TABLE w1830179_studio
(
    w1830179_studioId INTEGER,
    w1830179_studioName VARCHAR(50) UNIQUE NOT NULL,
    w1830179_studioAddress VARCHAR(150) UNIQUE NOT NULL,
    w1830179_studioPCode INTEGER NOT NULL,
    w1830179_studioTelNo VARCHAR(10)UNIQUE NOT NULL,
    PRIMARY KEY (w1830179_studioId)
);

```

□ Bind parameters

Bookmark this SQL query:

Delimiter ; □ Show this query here again □ Retain query box □ Rollback when finished ☑ Enable foreign key checks Go

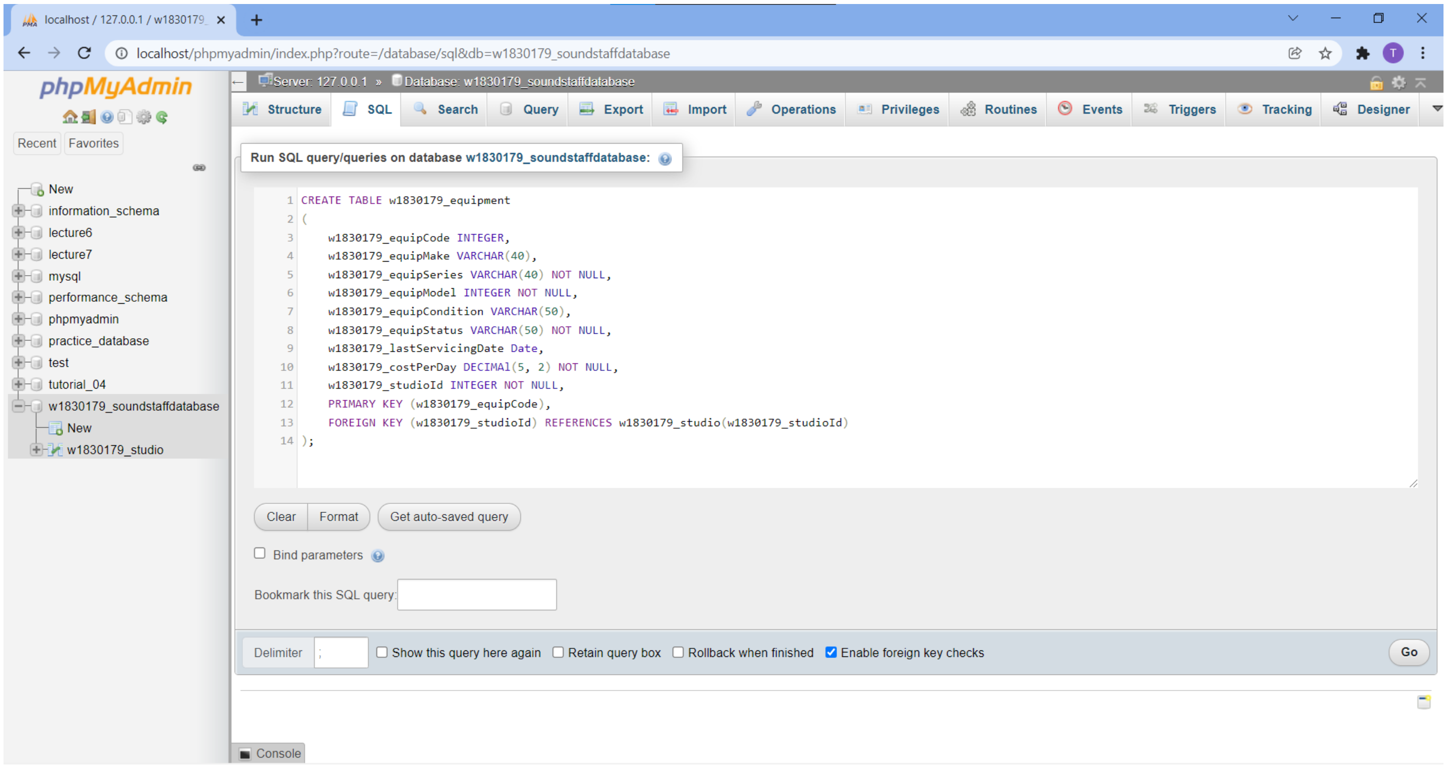


✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0115 seconds.)

```
CREATE TABLE w1830179_studio ( w1830179_studioId INTEGER, w1830179_studioName VARCHAR(50) UNIQUE NOT NULL, w1830179_studioAddress VARCHAR(150) UNIQUE NOT NULL,
w1830179_studioPCode INTEGER NOT NULL, w1830179_studioTelNo VARCHAR(10)UNIQUE NOT NULL, PRIMARY KEY (w1830179_studioId) );
```

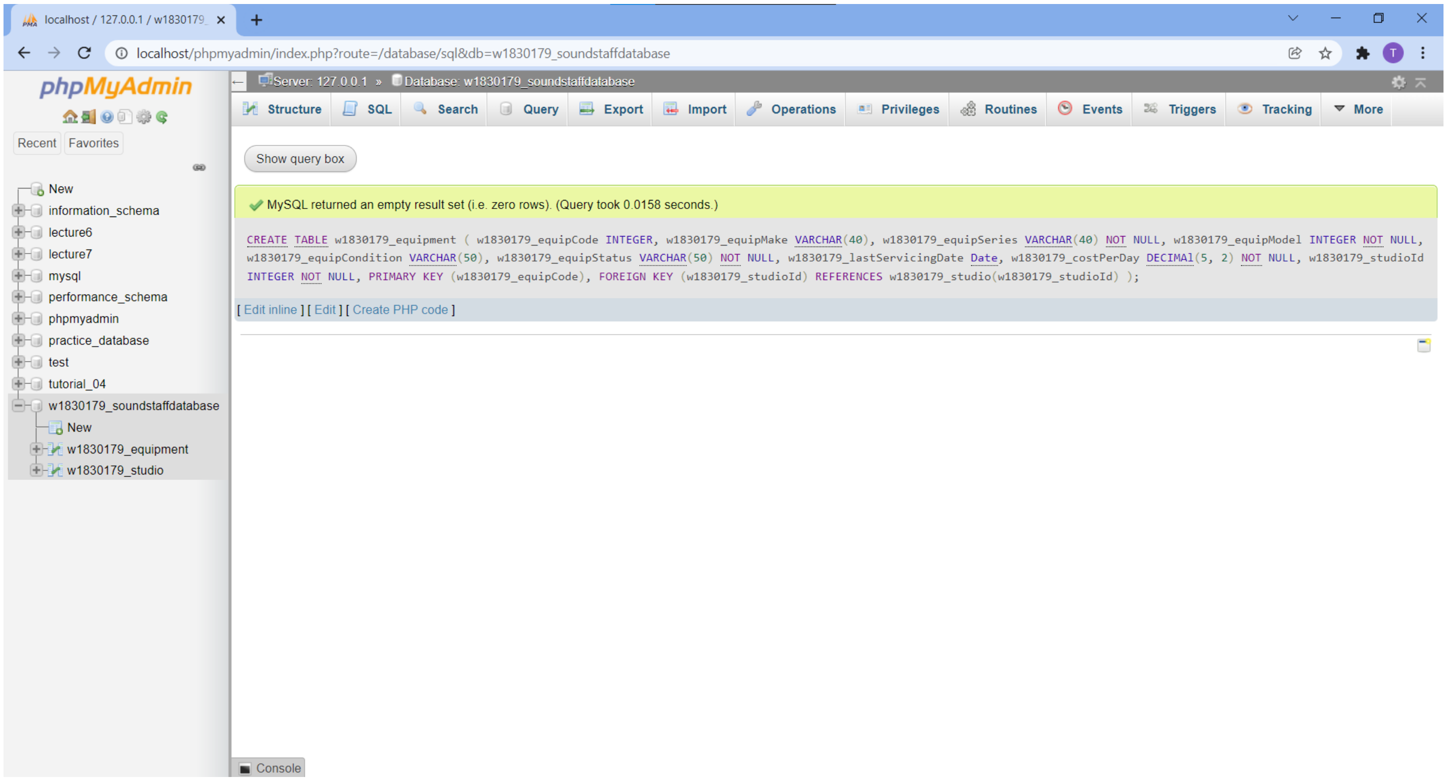
[Edit inline] [Edit] [Create PHP code]

Creating the equipment table



Run SQL query/queries on database w1830179_soundstaffdatabase:

```
CREATE TABLE w1830179_equipment
(
    w1830179_equipCode INTEGER,
    w1830179_equipMake VARCHAR(40),
    w1830179_equipSeries VARCHAR(40) NOT NULL,
    w1830179_equipModel INTEGER NOT NULL,
    w1830179_equipCondition VARCHAR(50),
    w1830179_equipStatus VARCHAR(50) NOT NULL,
    w1830179_lastServicingDate Date,
    w1830179_costPerDay DECIMAl(5, 2) NOT NULL,
    w1830179_studioId INTEGER NOT NULL,
    PRIMARY KEY (w1830179_equipCode),
    FOREIGN KEY (w1830179_studioId) REFERENCES w1830179_studio(w1830179_studioId)
);
```

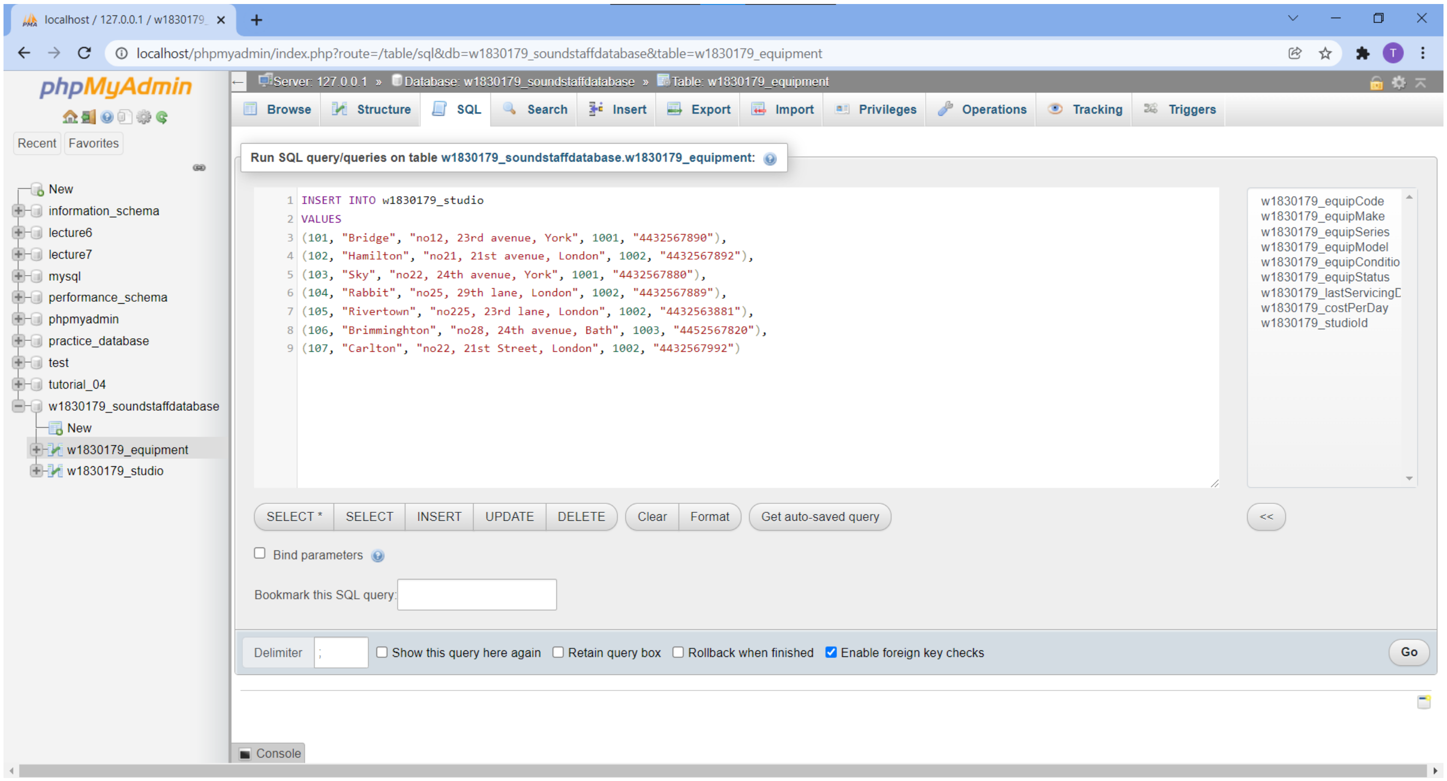


✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0158 seconds.)

```
CREATE TABLE w1830179_equipment ( w1830179_equipCode INTEGER, w1830179_equipMake VARCHAR(40), w1830179_equipSeries VARCHAR(40) NOT NULL, w1830179_equipModel INTEGER NOT NULL, w1830179_equipCondition VARCHAR(50), w1830179_equipStatus VARCHAR(50) NOT NULL, w1830179_lastServicingDate Date, w1830179_costPerDay DECIMAl(5, 2) NOT NULL, w1830179_studioId INTEGER NOT NULL, PRIMARY KEY (w1830179_equipCode), FOREIGN KEY (w1830179_studioId) REFERENCES w1830179_studio(w1830179_studioId) );
```

[Edit inline] [Edit] [Create PHP code]

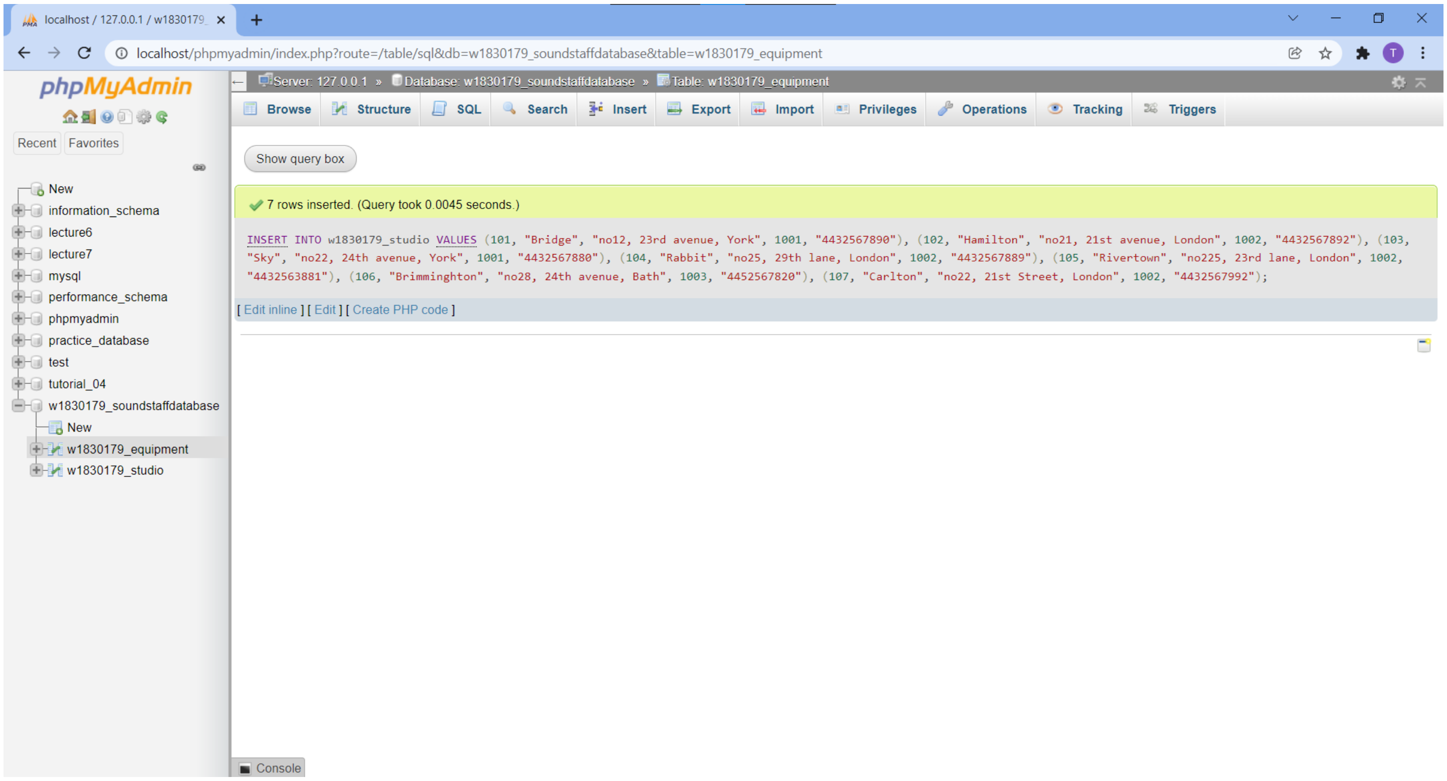
Adding data into studio table



Run SQL query/queries on table w1830179_soundstaffdatabase.w1830179_equipment:

```
INSERT INTO w1830179_studio
VALUES
(101, "Bridge", "no12, 23rd avenue, York", 1001, "4432567890"),
(102, "Hamilton", "no21, 21st avenue, London", 1002, "4432567892"),
(103, "Sky", "no22, 24th avenue, York", 1001, "4432567880"),
(104, "Rabbit", "no25, 29th lane, London", 1002, "4432567889"),
(105, "Rivertown", "no225, 23rd lane, London", 1002, "4432563881"),
(106, "Brimminghton", "no28, 24th avenue, Bath", 1003, "4452567820"),
(107, "Carlton", "no22, 21st Street, London", 1002, "4432567992")
```

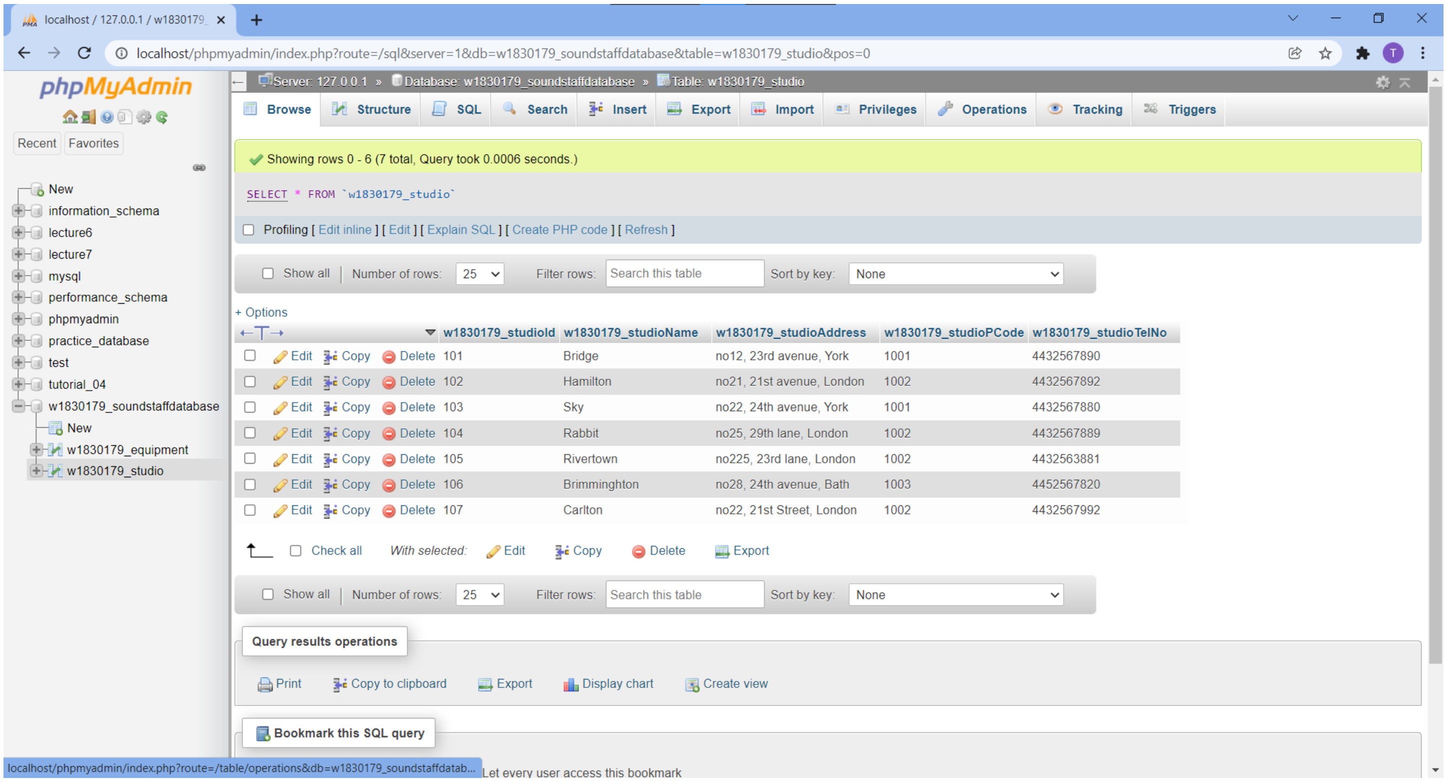
localhost / 127.0.0.1 / w1830179
localhost/phpmyadmin/index.php?route=/table/sql&db=w1830179_soundstaffdatabase&table=w1830179_equipment
phpMyAdmin
Recent Favorites
New
information_schema
lecture6
lecture7
mysql
performance_schema
phpmyadmin
practice_database
test
tutorial_04
w1830179_soundstaffdatabase
New
w1830179_equipment
w1830179_studio
Server: 127.0.0.1 » Database: w1830179_soundstaffdatabase » Table: w1830179_equipment
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
w1830179_equipCode
w1830179_equipMake
w1830179_equipSeries
w1830179_equipModel
w1830179_equipConditio
w1830179_equipStatus
w1830179_lastServicingD
w1830179_costPerDay
w1830179_studioId
SELECT * SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query
<<
Bind parameters
Bookmark this SQL query:
Delimiter ;
Show this query here again
Retain query box
Rollback when finished
Enable foreign key checks
Go
Console



✓ 7 rows inserted. (Query took 0.0045 seconds.)

```
INSERT INTO w1830179_studio VALUES (101, "Bridge", "no12, 23rd avenue, York", 1001, "4432567890"), (102, "Hamilton", "no21, 21st avenue, London", 1002, "4432567892"), (103, "Sky", "no22, 24th avenue, York", 1001, "4432567880"), (104, "Rabbit", "no25, 29th lane, London", 1002, "4432567889"), (105, "Rivertown", "no225, 23rd lane, London", 1002, "4432563881"), (106, "Brimminghton", "no28, 24th avenue, Bath", 1003, "4452567820"), (107, "Carlton", "no22, 21st Street, London", 1002, "4432567992");
```

localhost / 127.0.0.1 / w1830179
localhost/phpmyadmin/index.php?route=/table/sql&db=w1830179_soundstaffdatabase&table=w1830179_equipment
phpMyAdmin
Recent Favorites
New
information_schema
lecture6
lecture7
mysql
performance_schema
phpmyadmin
practice_database
test
tutorial_04
w1830179_soundstaffdatabase
New
w1830179_equipment
w1830179_studio
Server: 127.0.0.1 » Database: w1830179_soundstaffdatabase » Table: w1830179_equipment
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
Show query box
[Edit inline] [Edit] [Create PHP code]
Console



localhost / 127.0.0.1 / w1830179
localhost/phpmyadmin/index.php?route=/sql&server=1&db=w1830179_soundstaffdatabase&table=w1830179_studio&pos=0
phpMyAdmin
Recent
Favorites
New
information_schema
lecture6
lecture7
mysql
performance_schema
phpmyadmin
practice_database
test
tutorial_04
w1830179_soundstaffdatabase
New
w1830179_equipment
w1830179_studio
Server: 127.0.0.1 » Database: w1830179_soundstaffdatabase » Table: w1830179_studio
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers

✓ Showing rows 0 - 6 (7 total, Query took 0.0006 seconds.)

```
SELECT * FROM `w1830179_studio`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

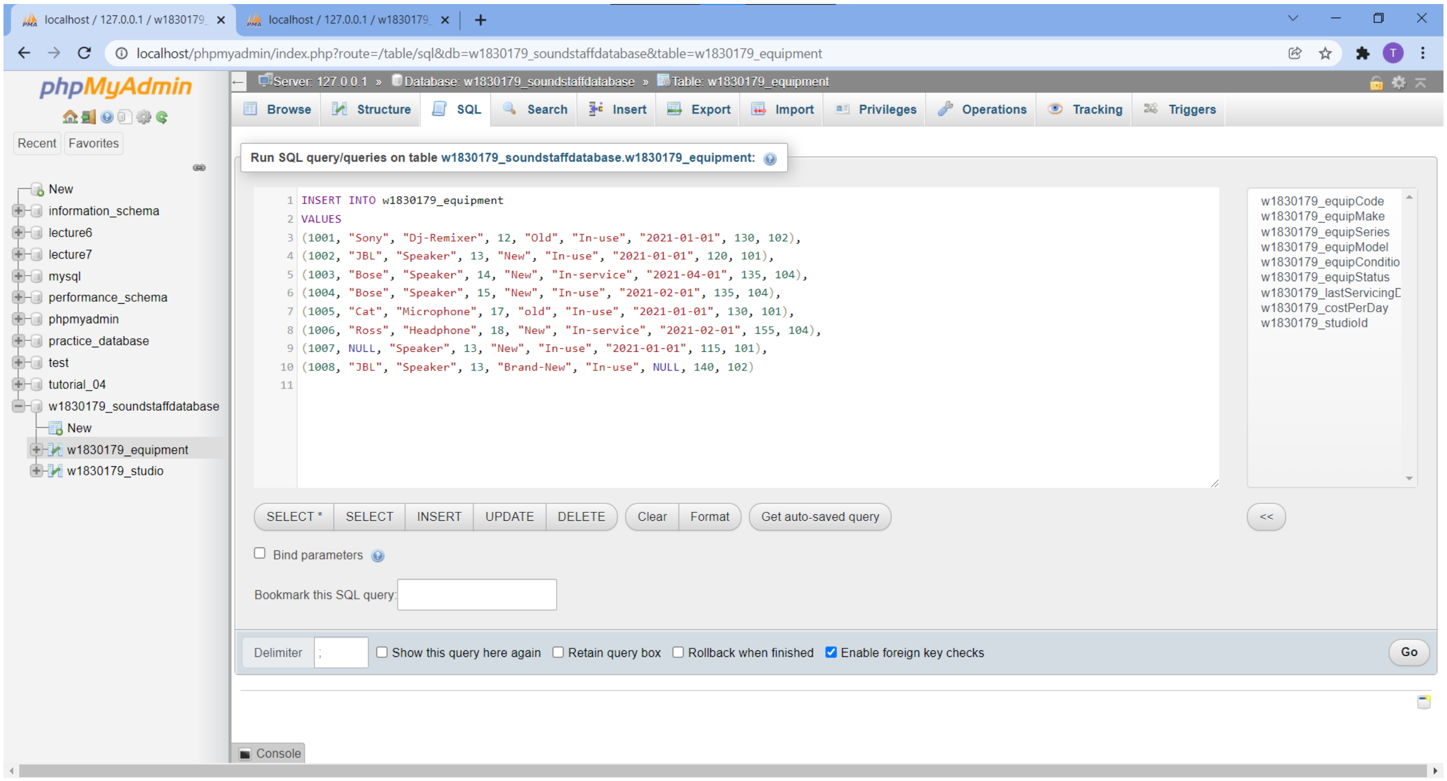
	w1830179_studioId	w1830179_studioName	w1830179_studioAddress	w1830179_studioPCode	w1830179_studioTelNo
□ Edit Copy Delete	101	Bridge	no12, 23rd avenue, York	1001	4432567890
□ Edit Copy Delete	102	Hamilton	no21, 21st avenue, London	1002	4432567892
□ Edit Copy Delete	103	Sky	no22, 24th avenue, York	1001	4432567880
□ Edit Copy Delete	104	Rabbit	no25, 29th lane, London	1002	4432567889
□ Edit Copy Delete	105	Rivertown	no225, 23rd lane, London	1002	4432563881
□ Edit Copy Delete	106	Brimminghton	no28, 24th avenue, Bath	1003	4452567820
□ Edit Copy Delete	107	Carlton	no22, 21st Street, London	1002	4432567992

□ Check all *With selected:* Edit Copy Delete Export

□ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations
Print
Copy to clipboard
Export
Display chart
Create view
Bookmark this SQL query
localhost/phpmyadmin/index.php?route=/table/operations&db=w1830179_soundstaffdatab...
Let every user access this bookmark

Adding data into equipment table

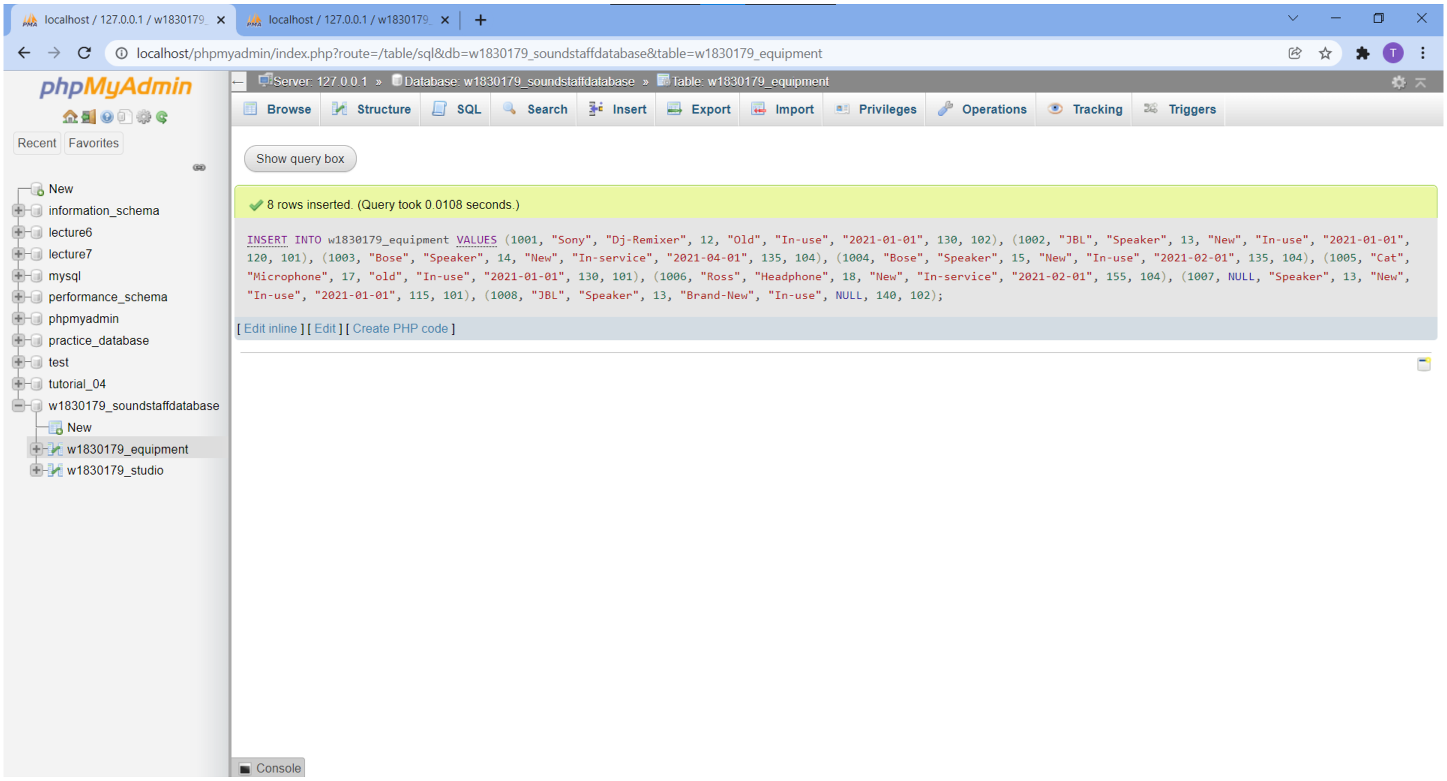


localhost/phpmyadmin/index.php?route=/table/sql&db=w1830179_soundstaffdatabase&table=w1830179_equipment

Run SQL query/queries on table w1830179_soundstaffdatabase.w1830179_equipment:

```sql
INSERT INTO w1830179_equipment
VALUES
(1001, "Sony", "Dj-Remixer", 12, "Old", "In-use", "2021-01-01", 130, 102),
(1002, "JBL", "Speaker", 13, "New", "In-use", "2021-01-01", 120, 101),
(1003, "Bose", "Speaker", 14, "New", "In-service", "2021-04-01", 135, 104),
(1004, "Bose", "Speaker", 15, "New", "In-use", "2021-02-01", 135, 104),
(1005, "Cat", "Microphone", 17, "old", "In-use", "2021-01-01", 130, 101),
(1006, "Ross", "Headphone", 18, "New", "In-service", "2021-02-01", 155, 104),
(1007, NULL, "Speaker", 13, "New", "In-use", "2021-01-01", 115, 101),
(1008, "JBL", "Speaker", 13, "Brand-New", "In-use", NULL, 140, 102)
```

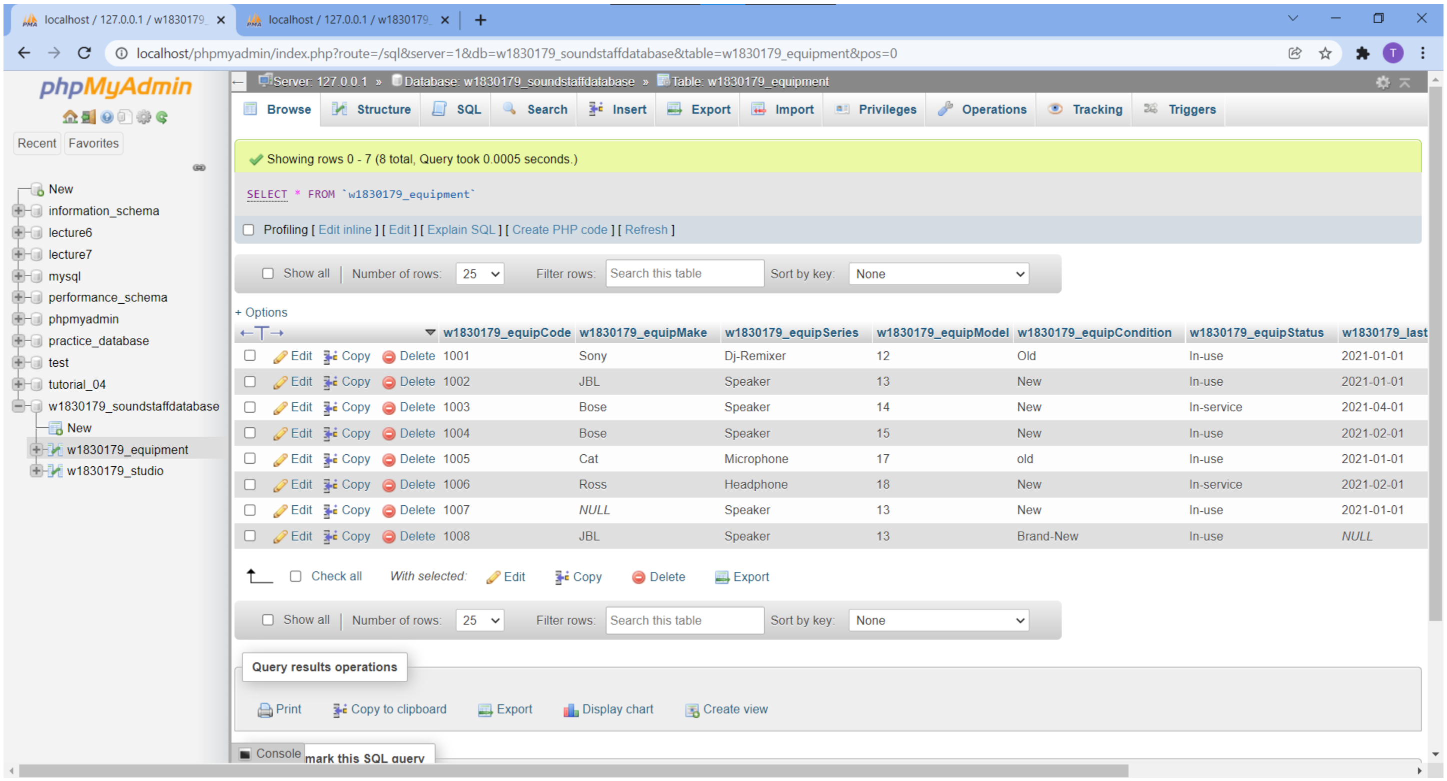
□ Show this query here again □ Retain query box □ Rollback when finished ☑ Enable foreign key checks



localhost/phpmyadmin/index.php?route=/table/sql&db=w1830179_soundstaffdatabase&table=w1830179_equipment

✓ 8 rows inserted. (Query took 0.0108 seconds.)

```sql
INSERT INTO w1830179_equipment VALUES (1001, "Sony", "Dj-Remixer", 12, "Old", "In-use", "2021-01-01", 130, 102), (1002, "JBL", "Speaker", 13, "New", "In-use", "2021-01-01", 120, 101), (1003, "Bose", "Speaker", 14, "New", "In-service", "2021-04-01", 135, 104), (1004, "Bose", "Speaker", 15, "New", "In-use", "2021-02-01", 135, 104), (1005, "Cat", "Microphone", 17, "old", "In-use", "2021-01-01", 130, 101), (1006, "Ross", "Headphone", 18, "New", "In-service", "2021-02-01", 155, 104), (1007, NULL, "Speaker", 13, "New", "In-use", "2021-01-01", 115, 101), (1008, "JBL", "Speaker", 13, "Brand-New", "In-use", NULL, 140, 102);
```



localhost/phpmyadmin/index.php?route=/sql&server=1&db=w1830179_soundstaffdatabase&table=w1830179_equipment&pos=0

Server: 127.0.0.1 » Database: w1830179_soundstaffdatabase » Table: w1830179_equipment

✓ Showing rows 0 - 7 (8 total, Query took 0.0005 seconds.)

```
SELECT * FROM `w1830179_equipment`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	w1830179_equipCode	w1830179_equipMake	w1830179_equipSeries	w1830179_equipModel	w1830179_equipCondition	w1830179_equipStatus	w1830179_last
□ Edit Copy Delete	1001	Sony	Dj-Remixer	12	Old	In-use	2021-01-01
□ Edit Copy Delete	1002	JBL	Speaker	13	New	In-use	2021-01-01
□ Edit Copy Delete	1003	Bose	Speaker	14	New	In-service	2021-04-01
□ Edit Copy Delete	1004	Bose	Speaker	15	New	In-use	2021-02-01
□ Edit Copy Delete	1005	Cat	Microphone	17	old	In-use	2021-01-01
□ Edit Copy Delete	1006	Ross	Headphone	18	New	In-service	2021-02-01
□ Edit Copy Delete	1007	*NULL*	Speaker	13	New	In-use	2021-01-01
□ Edit Copy Delete	1008	JBL	Speaker	13	Brand-New	In-use	*NULL*

□ Check all *With selected:* Edit Copy Delete Export

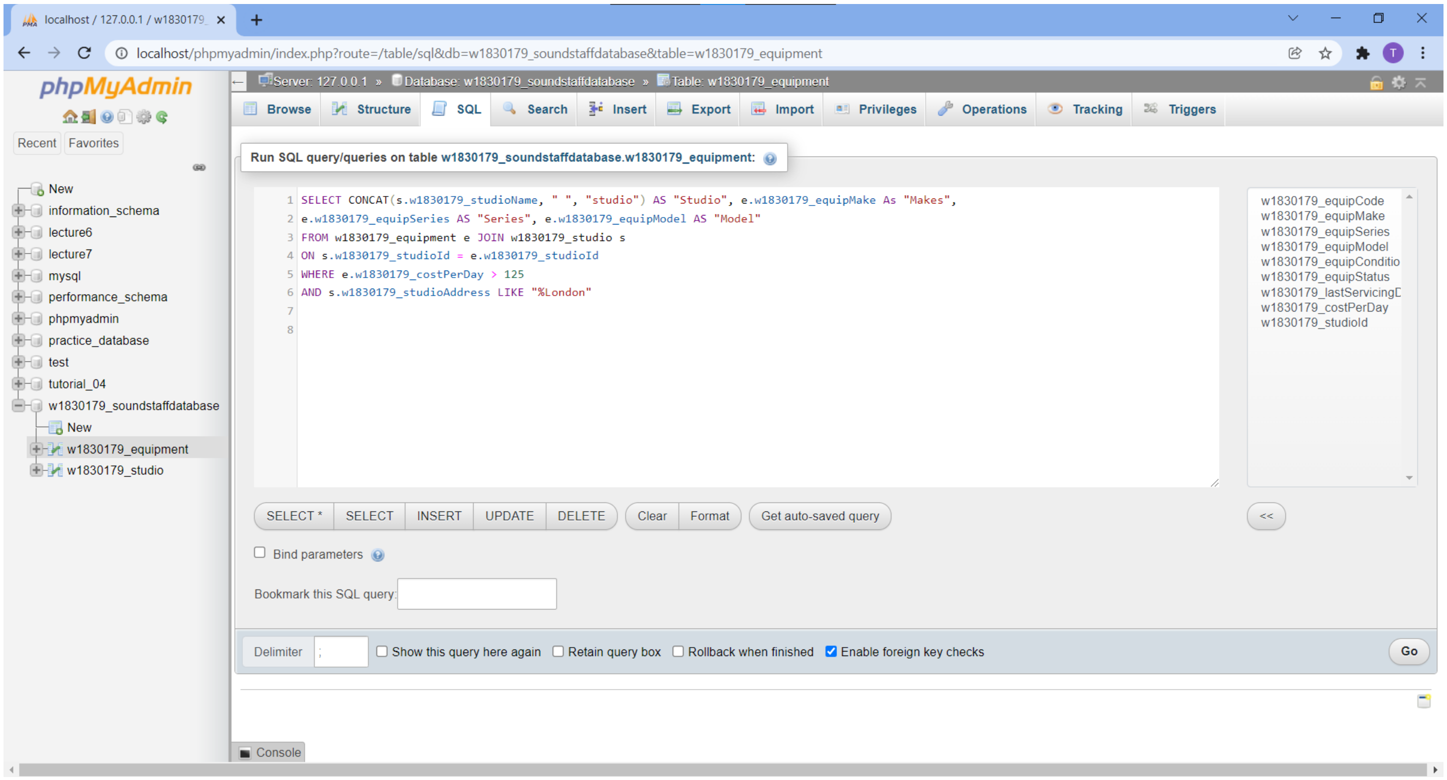
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Console

Answer



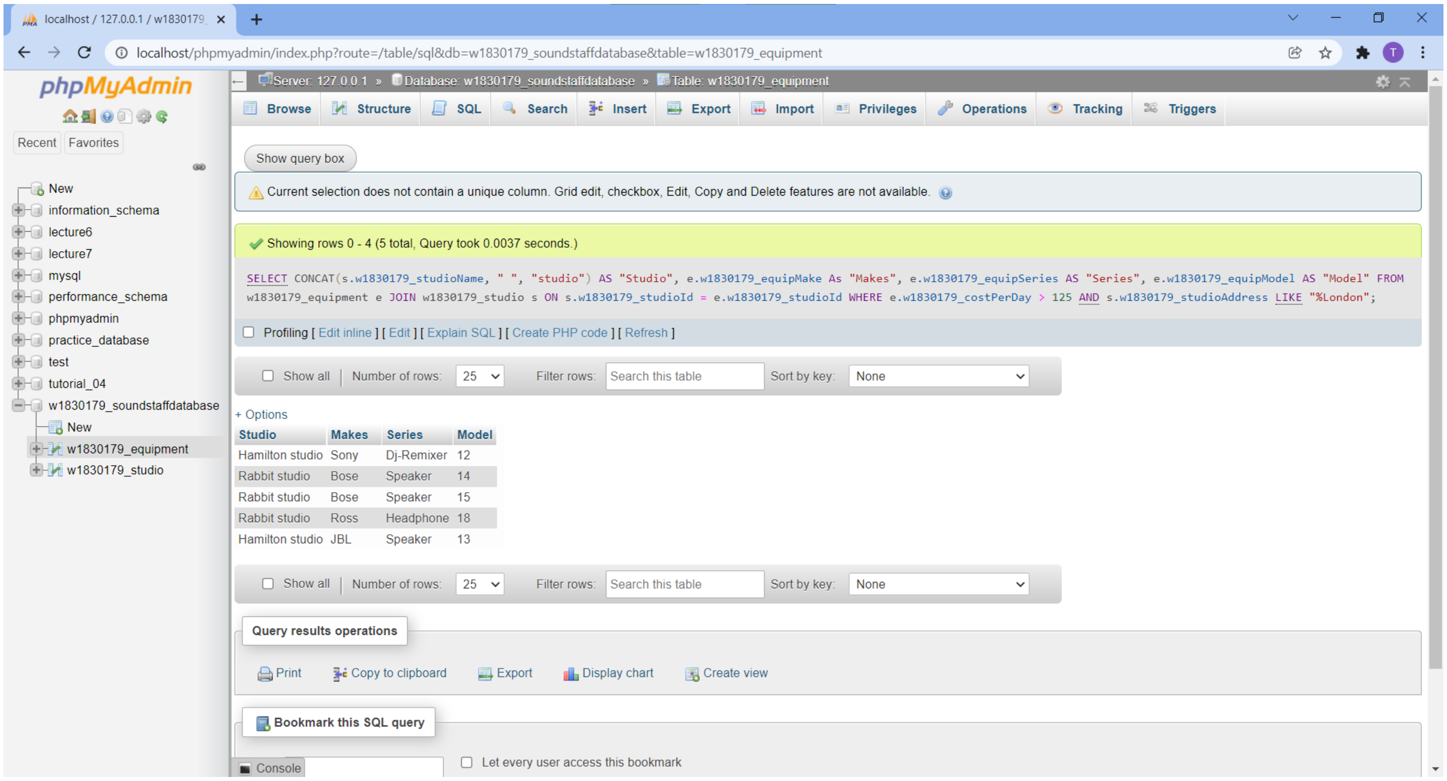
Run SQL query/queries on table w1830179_soundstaffdatabase.w1830179_equipment:

```
SELECT CONCAT(s.w1830179_studioName, " ", "studio") AS "Studio", e.w1830179_equipMake As "Makes",
e.w1830179_equipSeries AS "Series", e.w1830179_equipModel AS "Model"
FROM w1830179_equipment e JOIN w1830179_studio s
ON s.w1830179_studioId = e.w1830179_studioId
WHERE e.w1830179_costPerDay > 125
AND s.w1830179_studioAddress LIKE "%London"
```

□ Bind parameters

Bookmark this SQL query:

Delimiter ; □ Show this query here again □ Retain query box □ Rollback when finished ☑ Enable foreign key checks



Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 4 (5 total, Query took 0.0037 seconds.)

```
SELECT CONCAT(s.w1830179_studioName, " ", "studio") AS "Studio", e.w1830179_equipMake As "Makes", e.w1830179_equipSeries AS "Series", e.w1830179_equipModel AS "Model" FROM
w1830179_equipment e JOIN w1830179_studio s ON s.w1830179_studioId = e.w1830179_studioId WHERE e.w1830179_costPerDay > 125 AND s.w1830179_studioAddress LIKE "%London";
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

Studio	Makes	Series	Model
Hamilton studio	Sony	Dj-Remixer	12
Rabbit studio	Bose	Speaker	14
Rabbit studio	Bose	Speaker	15
Rabbit studio	Ross	Headphone	18
Hamilton studio	JBL	Speaker	13

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Bookmark this SQL query

□ Let every user access this bookmark

7) Comparative analysis table Between Relational Databases vs. NoSQL databases with a view to inform the decision-making of the management of a firm.

Before Getting into the comparative analysis table Its Important to know what relational database is and what is NoSQL database.

For years, relational databases have been a widely used technology. They are well-established, well-proven, and widely used. There are several database products, tools, and expertise to choose from. Relational databases keep track of data tables that are connected to one another. These tables have a set schema, manage data with SQL (Structured Query Language), and provide ACID guarantees.(Robvet, 2021)

No-SQL databases refer to high-performance, non-relational data stores. They shine out for their ease of use, flexibility, resilience, and availability. NoSQL stores unstructured or semi-structured data, in key-value pairs or JSON documents, rather than joining tables of normalized data. Beyond the scope of a single database partition, most No-SQL databases do not satisfy ACID guarantees. NoSQL datastores are preferred by high-volume services that demand sub-second response times. (Robvet, 2021)

This will give you an idea about these database systems. Now Using a comparative analysis table, we will get into more details.

Criterion	Relational Databases	NoSQL databases
1. schemas	Before you can insert data, you must first define your schema. Changing the schema structure of a relational database is costly, time-consuming, and frequently involves downtime or service interruptions. (Mohamed, Altrafi and Ismail, 2014)	Have a dynamic schema and work with "unstructured data." This implies you can start developing your application before even defining the schema. NoSQL databases are easier to modify when data and requirements change since they do not require a predefined schema. (Mohamed, Altrafi and Ismail, 2014)
2. Structure	organize and store data by tables with fixed columns and rows. (Berga and Franco, 2021)	can be graph, document-oriented, key-value, column-oriented, and others. (Berga and Franco, 2021)
3.Data Consistency	Relational Databases are great at enforcing consistency. (Robvet, 2021)	It relies on the database management system, as some offer high consistency while others merely provide eventual consistency. (Robvet, 2021)

4. Storage Type	Highly Available Storage (SAN, RAID, etc.). (Li and Manoharan, 2013)	Commodity drives storage (standard HDDs, JBOD). (Li and Manoharan, 2013)
5. infrastructures	Relational Databases often require high-end special-purpose hardware. (Berga and Franco, 2021)	NoSQL Databases are well known for their ability to run using cheap general-purpose hardware and scale horizontally. (Berga and Franco, 2021)
6. Confidentiality	Since rdbms use encryption techniques to store data encrypted, data confidentiality is frequently accomplished. (Mohamed, Altrafi and Ismail, 2014)	Because data is frequently stored in clear form, data confidentiality is not achieved. (Mohamed, Altrafi and Ismail, 2014)
7. Data Integrity	The ACID properties, that are used in relational databases, ensure that database transactions are processed securely and that data is integrated. (Mohamed, Altrafi and Ismail, 2014)	Since one of the BASE properties principle is that data integrity is not always achieved in NoSQL databases, integrity of data is not always maintained in NoSQL databases. (Mohamed, Altrafi and Ismail, 2014)
8. Auditing	Provide auditing techniques that allow writing to the database. (Mohamed, Altrafi and Ismail, 2014)	Most of NoSQL databases don’t provide auditing. (Mohamed, Altrafi and Ismail, 2014)

9.Cloud	They are not well suited for cloud environments because they do not offer comprehensive data search and scaling them beyond a certain point is hard. (Mohamed, Altrafi and Ismail, 2014)	Cloud databases are not ACID compliant, but they do provide better availability, scalability, performance, and flexibility. They can manage unstructured, semi structured, or structured data. NoSQL databases are particularly desirable for cloud computing because they have all the properties that make them ideal for this purpose. (Mohamed, Altrafi and Ismail, 2014)
10.Crash Recovery	Grantee crash recovery through the recovery manager, which oversees ensuring transactional atomicity and persistence through the usage of log files as well as the ARIES algorithm. (Mohamed, Altrafi and Ismail, 2014)	Rely on replication as a backup to recover from a crash, but some have other methods, such as Mango dB’s Journal file. (Mohamed, Altrafi and Ismail, 2014)
11.Using multiple Data types.	Relational database takes longer time to store data types such as images If using multiple data types this is not recommended .(Abdullah and Zhuge, 2015)	Much faster when using multiple data type specially using 1000 s of images. This shows a considerable time gap than the relational databases. (Mohamed, Altrafi and Ismail, 2014)

7.1) Citations

Abdullah, A. and Zhuge, Q. (2015). From Relational Databases to NoSQL Databases: Performance Evaluation. *Research Journal of Applied Sciences, Engineering and Technology*, 11 (4), 434–439. Available from https://doi.org/10.19026/rjaset.11.1799.

Berga, M. and Franco, T. (2021). SQL vs NoSQL: when to use? *Blog | Imaginary Cloud*. Available from https://www.imaginarycloud.com/blog/sql-vs-nosql/ [Accessed 3 December 2021].

Li, Y. and Manoharan, S. (2013). A performance comparison of SQL and NoSQL databases. *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*. August 2013. Victoria, BC, Canada: IEEE, 15–19. Available from https://doi.org/10.1109/PACRIM.2013.6625441 [Accessed 3 December 2021].

Mohamed, M.A., Altrafi, O.G. and Ismail, M.O. (2014). Relational vs. NoSQL Databases: A Survey. 5 March 2014. 5.

Robvet. (2021). Relational vs. NoSQL data. Available from https://docs.microsoft.com/en-us/dotnet/architecture/cloud-native/relational-vs-nosql-data [Accessed 4 December 2021].