



National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Faculty of Computing

CS220: Database Systems

Class: BESE-13A

Lab 07: SQL Joins

Date: November 30th, 2023

Time: 2:00 PM-5:00 PM

Instructor: Dr. Shah Khalid

Lab Engineer: Sundas Dawood

Name: Aimen Munawar
Class: BESE-13-A
CMS ID: 415867



Lab 07: SQL Joins

Introduction

A SQL JOIN clause combines records from two or more tables in a database. It creates a set that can be saved as a table or used as it is. A JOIN is a means for combining fields from two tables by using values common to both tables.

Objectives

After performing this lab students should be able to:

1. Design SQL queries to retrieve data from multiple tables by using JOIN operation.
2. Select among INNER, OUTER, NATURAL, CROSS join as and when required.

Tools/Software Requirement

- MySQL Community Server 5.6 or later
- MySQL Workbench 6.1 or later
- Sakila Database and Company Database

Description

1. This lab assumes that MySQL Community Server is running, Sakila database has been loaded using MySQL Workbench, and the query window is open.
2. Execute the JOIN queries on Sakila including cross join, inner join, natural join, outer join (left outer join, right outer join, and full outer join) and observe how each join operation differs from the other.
3. For your practice, create the following two tables and execute the following queries to better understand how each join operation works.
 - a. Specify a primary-foreign key relationship between both tables.
 - b. Produce a set of all users who are enrolled in a course.
 - c. List of all students and their courses even if they're not enrolled in any course yet.
 - d. List all records of both tables regardless of any match.

Table: Users

id	name	course_id
1	Alice	1
2	Bob	1



National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

3	Caroline	2
4	David	5
5	Emma	(NULL)

Table: Course

course_id	language
1	HTML5
2	NULL
3	JavaScript
4	PHP
5	MySQL

4. Try running the following SQL JOIN queries. You can try equivalent SQL queries for the same result as well.
- List rental and return date for the movie BREAKING HOME (notice one of the join uses **on** clause whereas the other uses **using** clause. 13 entries of renting this movie should be found from the database.

use sakila;

```
select r.rental_date, return_date  
from rental r join inventory i  
using (inventory_id) join film f  
on (f.film_id=i.film_id)  
where f.title like 'BREAKING HOME' ;
```

- List of movie titles that were never rented. There are 43 of them. Notice that there are two outer joins as the movie might be missing altogether from the inventory



National University of Sciences and Technology (NUST) School of Electrical Engineering and Computer Science

but may still be included in the result set. On the other hand, movies might be available in the inventory but never rented, the SQL query covers both cases.

```
select f.title
from film f left join inventory
i using (film_id) left join
rental r using (inventory_id)
where (i.inventory_id is null or r.rental_id is null);
```

Lab Task

Write SQL queries for the following information needs. You should execute your attempts on SQL workbench and make necessary corrections if needed using **Sakila database**.

- a. **What category does the movie CHOCOLATE DUCK belong to?**

Code:

```
select fm.title ,cat.name As Category
from film fm
join film_category fc on fm.film_id=fc.film_id
join category cat on cat.category_id=fc.category_id
where fm.title ='chocolate duck';
```

Output:



```
8
9 • select fm.title ,cat.name As Category
10 from film fm
11 join film_category fc on fm.film_id=fc.film_id
12 join category cat on cat.category_id=fc.category_id
13 where fm.title ='chocolate duck';
14
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	title	Category
▶	CHOCOLATE DUCK	Foreign

Result 57 x

Output				
Action Output				
#	Time	Action	Message	
✓ 70	16:00:49	SELECT fm.title ,rt.rental_id,rt.rental_date,DATEDIFF(now(), rt.rental_date) AS rental_duration FROM rental rt JOIN inventory it ON it.inventory_...	183 row(s) returned	
✓ 71	16:01:00	SELECT fm.title ,rt.rental_id,rt.rental_date,DATEDIFF(rt.return_date, rt.rental_date) AS rental_duration FROM rental rt JOIN inventory it ON it.in...	183 row(s) returned	
✓ 72	16:02:19	select fm.title ,cat.name As Category from film fm join film_category fc on fm.film_id=fc.film_id join category cat on cat.category_id=fc.category_id...	1 row(s) returned	

b. Track the location (city & country) of staff member Jon.

Code:

```
select first_name, city ,country
from staff st
join address ad on st.address_id=ad.address_id
join city ct on ct.city_id=ad.city_id
join country co on co.country_id=ct.country_id
where st.first_name ='Jon';
```

Output:



```
8 • select first_name, city ,country
9   from staff st
10  join address ad on st.address_id=ad.address_id
11  join city ct on ct.city_id=ad.city_id
12  join country co on co.country_id=ct.country_id
13  where st.first_name = 'Jon';
14
```

Result Grid

Filter Rows: Export: Wrap Cell Content:

	first_name	city	country
▶	Jon	Woodridge	Australia

Result 66 x

Output

Action Output

#	Time	Action	Message
✓ 79	16:03:31	select count(*) as Films_Number from film where title like "%crocodile%" AND "%shark%" LIMIT 0, 1000	1 row(s) returned
✓ 80	16:03:31	SELECT language_id FROM language WHERE name = 'English' OR name = 'French' LIMIT 0, 1000	2 row(s) returned
✓ 81	16:03:34	select first_name, city ,country from staff st join address ad on st.address_id=ad.address_id join city ct on ct.city_id=ad.city_id join country co o...	1 row(s) returned

c. Retrieve first and last name of actors who played in ALONE TRIP.

Code:

```
select first_name,last_name,fm.title
from actor aa
join film_actor fa on fa.actor_id=aa.actor_id
join film fm on fm.film_id=fa.film_id
where title ="alone trip";
```

Output:



```
13 • select first_name,last_name,fm.title
14 from actor aa
15 join film_actor fa on fa.actor_id=aa.actor_id
16 join film fm on fm.film_id=fa.film_id
17 where title ="alone trip";
18
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: 12

	first_name	last_name	title
▶	ED	CHASE	ALONE TRIP
	KARL	BERRY	ALONE TRIP
	UMA	WOOD	ALONE TRIP
	WOODY	JOLIE	ALONE TRIP
	SPENCER	DEPP	ALONE TRIP
	CHRIS	DEPP	ALONE TRIP
	LAURENCE	BULLOCK	ALONE TRIP
	RENEE	BALL	ALONE TRIP

Result 67 x

Output

Action Output

#	Time	Action	Message
✓ 80	16:03:31	SELECT language_id FROM language WHERE name = 'English' OR name = 'French' LIMIT 0, 1000	2 row(s) returned
✓ 81	16:03:34	select first_name, city ,country from staff st join address ad on st.address_id=ad.address_id join city ct on ct.city_id=ad.city_id join country co o...	1 row(s) returned
✓ 82	16:04:14	select first_name,last_name,fm.title from actor aa join film_actor fa on fa.actor_id=aa.actor_id join film fm on fm.film_id=fa.film_id where title ="alo...	8 row(s) returned

d. List of movies in Games category having rental rate of more than \$4 and sorted on movie titles.

Code:

```
select title
from film fm
join film_category fc on fm.film_id=fc.film_id
join category cat on cat.category_id=fc.category_id
WHERE cat.name = 'Games' AND fm.rental_rate > 4
order by fm.title;
```

Output:



```
18 • select title
19   from film fm
20  join film_category fc on fm.film_id=fc.film_id
21  join category cat on cat.category_id=fc.category_id
22  where cat.name = 'Games' AND fm.rental_rate > 4
23  order by fm.title;
24
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	title
▶	AUTUMN CROW
	CANDLES GRAPES
	CHICAGO NORTH
	DARLING BREAKING
	DAWN POND
	DAZED PUNK
	DETAILS PACKER
	FEVER EMPIRE
	FIRE WOLVES
	GRINCH MESSAGE
	HEAD STRANGER
	LAMBS CINCINNATI
	MALTESE HOPE
	MASSACRE USUAL
	MONSOON CAUSE
	MOONSHINE CABIN
	MOONWALKER FOOL
	NAME DETECTIVE
	NIGHTMARE CHILL
	PANKY SUBMARINE

Result 68 x

Output

Action Output

#	Time	Action	Message
81	16:03:34	select first_name, city .country from staff st join address ad on st.address_id=ad.address_id join city ct on ct.city_id=ad.city_id join country co o...	1 row(s) returned
82	16:04:14	select first_name,last_name,fm.title from actor aa join film_actor fa on fa.actor_id=aa.actor_id join film fm on fm.film_id=fa.film_id where title ="alo...	8 row(s) returned
83	16:05:06	select title from film fm join film_category fc on fm.film_id=fc.film_id join category cat on cat.category_id=fc.category_id WHERE cat.name = 'Ga...	27 row(s) returned

- e. **Email addresses of customers who rented a movie but didn't pay anything, include movie title in the output.**

Code:

```
SELECT ct.email, fm.title
FROM customer ct
JOIN rental rt ON rt.customer_id = ct.customer_id
JOIN payment pm ON pm.rental_id = rt.rental_id
JOIN inventory it ON rt.inventory_id = it.inventory_id
JOIN film fm ON it.film_id = fm.film_id
WHERE pm.amount = 0;
```

Output:



```
24 • SELECT ct.email, fm.title
25 FROM customer ct
26 JOIN rental rt ON rt.customer_id = ct.customer_id
27 JOIN payment pm ON pm.rental_id = rt.rental_id
28 JOIN inventory it ON rt.inventory_id = it.inventory_id
29 JOIN film fm ON it.film_id = fm.film_id
30 WHERE pm.amount = 0;
31
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

email	title
HELEN.HARRIS@sakilacustomer.org	SMOKING BARBARELLA
CAROLYN.PEREZ@sakilacustomer.org	RIVER OUTLAW
CHRISTINE.ROBERTS@sakilacustomer.org	MOTIONS DETAILS
HEATHER.MORRIS@sakilacustomer.org	LAWLESS VISION
MILDRED.BAILEY@sakilacustomer.org	CHAMBER ITALIAN
TAMMY.SANDERS@sakilacustomer.org	TROUBLE DATE
TAMMY.SANDERS@sakilacustomer.org	SLEEPY JAPANESE
FLORENCE.WOODS@sakilacustomer.org	BLADE POLISH
GAIL.KNIGHT@sakilacustomer.org	CHASING FIGHT
CATHY.SPENCER@sakilacustomer.org	HOLES BRANNIGAN
ANNETTE.OLSON@sakilacustomer.org	DEER VIRGINIAN
LUCY.WHEELER@sakilacustomer.org	CURTAIN VIDEOTAPE
NATALIE.MEYER@sakilacustomer.org	WOMEN DORADO
ALLISON.STANLEY@sakilacustomer.org	CYCLONE FAMILY
MARGIE.WADE@sakilacustomer.org	PHILADELPHIA WIFE
CASSANDRA.WALTERS@sakilacustomer.org	PRINCESS GIANT
SONIA.GREGORY@sakilacustomer.org	FRIDA SLIPPER
JUSTIN.NGO@sakilacustomer.org	TITANIC BOONDOCK
LAWRENCE.LAWTON@sakilacustomer.org	HALF OUTFIELD
MIGUEL.BETANCOURT@sakilacustomer.org	STATE WASTELAND

Result 69 x

Output

Action Output

#	Time	Action	Message
82	16:04:14	select first_name,last_name,fm.title from actor aa join film_actor fa on fa.actor_id=aa.actor_id join film fm on fm.film_id=fa.film_id where title = "alo..."	8 row(s) returned
83	16:05:06	select title from film fm join film_category fc on fm.film_id=fc.film_id join category cat on cat.category_id=fc.category_id WHERE cat.name = 'Ga...	27 row(s) returned
84	16:05:46	SELECT ct.email, fm.title FROM customer ct JOIN rental rt ON rt.customer_id = ct.customer_id JOIN payment pm ON pm.rental_id = rt.rental_id ...	24 row(s) returned

f. **List of unpaid rentals (film title, rental id, rental date and for how many days rented). List should be sorted on film title and rental date.**

Code:

```
SELECT fm.title ,rt.rental_id,rt.rental_date,DATEDIFF(now(),
rt.rental_date) AS rental_duration
FROM rental rt
JOIN inventory it ON rt.inventory_id = it.inventory_id
JOIN film fm ON it.film_id = fm.film_id
WHERE rt.return_date IS NULL
ORDER BY fm.title, rt.rental_date;
```

Output:



```
31 • SELECT fm.title ,rt.rental_id,rt.rental_date,DATEDIFF(now(), rt.rental_date) AS rental_duration
32 FROM rental rt
33 JOIN inventory it ON rt.inventory_id = it.inventory_id
34 JOIN film fm ON it.film_id = fm.film_id
35 WHERE rt.return_date IS NULL
36 ORDER BY fm.title, rt.rental_date;
37
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
title	rental_id	rental_date	rental_duration
ACADEMY DINOSAUR	14098	2005-08-21 00:30:32	6644
ACE GOLDFINGER	13421	2006-02-14 15:16:03	6467
AFFAIR PREJUDICE	15542	2006-02-14 15:16:03	6467
AFRICAN EGG	15458	2006-02-14 15:16:03	6467
ALI FOREVER	15294	2006-02-14 15:16:03	6467
ALONE TRIP	12988	2006-02-14 15:16:03	6467
AMADEUS HOLY	12786	2006-02-14 15:16:03	6467
AMERICAN CROCUS	13478	2006-02-14 15:16:03	6467

Result 1 x

Output

Action Output

#	Time	Action	Message
1	16:32:39	SELECT fm.title ,rt.rental_id,rt.rental_date,DATEDIFF(now(), rt.rental_date) AS rental_duration FROM re...	183 row(s) returned

g. How many films involve the word “Crocodile” and a “Shark”?

Code:

```
select count(*) as Films_Number
from film
where title like '%crocodile%' AND "%shark%";
```

\



National University of Sciences and Technology (NUST) School of Electrical Engineering and Computer Science

Output:

```
38 • select count(*) as Films_Number
39 from film
40 where title like '%crocodile%' AND "%shark%";
41
42 • SELECT language_id
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Films_Number			
0			

Result 71 x			
Output			
Action Output			
#	Time	Action	Message
84	16:05:46	SELECT ct.email, fm.title FROM customer ct JOIN rental rt ON rt.customer_id = ct.customer_id JOIN payment pm ON pm.rental_id = rt.rental_id ...	24 row(s) returned
85	16:06:19	SELECT fm.title, rt.rental_id, rt.rental_date, DATEDIFF(rt.return_date, rt.rental_date) AS rental_duration FROM rental rt JOIN inventory it ON it.in...	183 row(s) returned
86	16:07:21	select count(*) as Films_Number from film where title like "%crocodile%" AND "%shark%" LIMIT 0, 1000	1 row(s) returned

h. Retrieve language ids of languages ENGLISH and FRENCH.

Code:

```
SELECT language_id
FROM language
WHERE name = 'English' OR name = 'French';
```

Output:



```
42 • SELECT language_id
43 FROM language
44 WHERE name = 'English' OR name = 'French';
45
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

language_id
1
5
NULL

language 72 x

Output

Action Output

#	Time	Action	Message
85	16:06:19	SELECT fm.title ,rt.rental_id,rt.rental_date,DATEDIFF(rt.return_date, rt.rental_date) AS rental_duration FROM rental rt JOIN inventory it ON it.in...	183 row(s) returned
86	16:07:21	select count(*) as Films_Number from film where title like "%crocodile%" AND "%shark%" LIMIT 0, 1000	1 row(s) returned
87	16:12:41	SELECT language_id FROM language WHERE name = 'English' OR name = 'French' LIMIT 0, 1000	2 row(s) returned

Deliverable

Submit a PDF document including the SQL queries to answer above-mentioned information as well as snapshot of their outcome when executed over MySQL using the Workbench.