

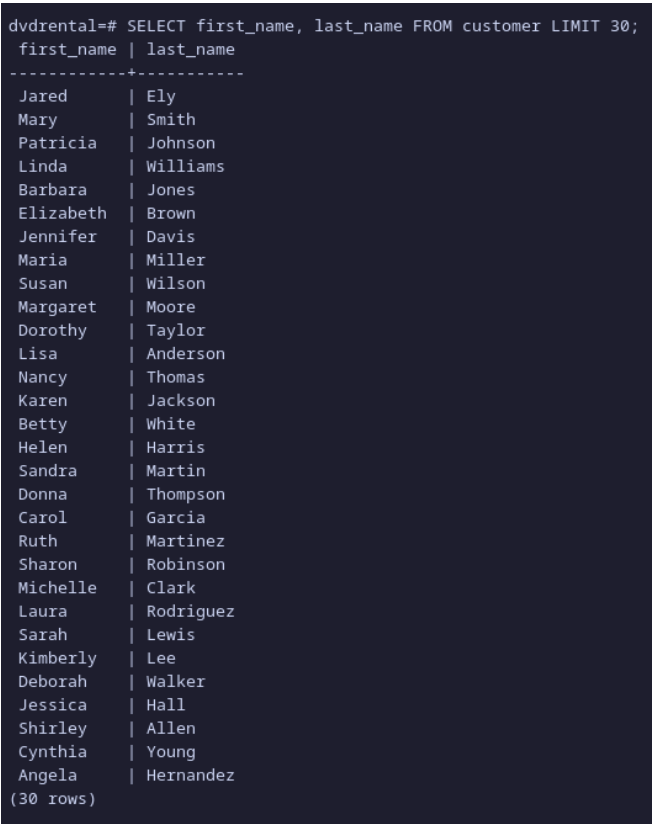
SEN-209: Designing Database Systems

Lab 5

Name: Thanawin Pattanaphol

Date: 29th September 2025

Basic Queries

SQL Query	Screenshot
<pre>SELECT first_name, last_name FROM customer LIMIT 30; (Limited to 30 rows for easy viewing)</pre>	 <pre>dvdrental=# SELECT first_name, last_name FROM customer LIMIT 30; first_name last_name -----+----- Jared Ely Mary Smith Patricia Johnson Linda Williams Barbara Jones Elizabeth Brown Jennifer Davis Maria Miller Susan Wilson Margaret Moore Dorothy Taylor Lisa Anderson Nancy Thomas Karen Jackson Betty White Helen Harris Sandra Martin Donna Thompson Carol Garcia Ruth Martinez Sharon Robinson Michelle Clark Laura Rodriguez Sarah Lewis Kimberly Lee Deborah Walker Jessica Hall Shirley Allen Cynthia Young Angela Hernandez (30 rows)</pre>

SEN-209: Designing Database Systems

Lab 5

Name: Thanawin Pattanaphol

Date: 29th September 2025

```
SELECT title FROM film LIMIT 30;  
(Query limited to 30 rows)
```

```
dvdrental=# SELECT title FROM film LIMIT 30;  
          title  
-----  
Chamber Italian  
Grosse Wonderful  
Airport Pollock  
Bright Encounters  
Academy Dinosaur  
Ace Goldfinger  
Adaptation Holes  
Affair Prejudice  
African Egg  
Agent Truman  
Airplane Sierra  
Alabama Devil  
Aladdin Calendar  
Alamo Videotape  
Alaska Phantom  
Date Speed  
Ali Forever  
Alice Fantasia  
Alien Center  
Alley Evolution  
Alone Trip  
Alter Victory  
Amadeus Holy  
Amelie Hellfighters  
American Circus  
Amistad Midsummer  
Anaconda Confessions  
Analyze Hoosiers  
Angels Life  
Annie Identity  
(30 rows)
```

```
SELECT DISTINCT rating FROM film;
```

```
dvdrental=# SELECT DISTINCT rating FROM film;  
          rating  
-----  
G  
PG-13  
PG  
R  
NC-17  
(5 rows)
```

SEN-209: Designing Database Systems

Lab 5

Name: Thanawin Pattanaphol

Date: 29th September 2025

<pre>SELECT title, rental_rate FROM film ORDER BY rental_rate ASC LIMIT 5;</pre>	<pre>dvdrental=# SELECT title, rental_rate FROM film title rental_rate -----+----- Alaska Phantom 0.99 Date Speed 0.99 Alamo Videotape 0.99 Academy Dinosaur 0.99 Alice Fantasia 0.99 (5 rows)</pre>
<pre>SELECT email FROM customer CU INNER JOIN address A ON A.address_id = CU.address_id INNER JOIN city C ON C.city_id = A.city_id WHERE C.city = 'London';</pre>	<pre>dvdrental=# SELECT email FROM customer CU INNER JOIN address A ON A.address_id = CU.address_id INNER JOIN city C ON C.city_id = A.city_id WHERE C.city = 'London'; email ----- mattie.hoffman@sakilacustomer.org cecil.vines@sakilacustomer.org (2 rows)</pre>

Intermediate Queries

SQL Query	Screenshot
<pre>SELECT COUNT(film_id) AS titles FROM film;</pre>	<pre>dvdrental=# SELECT COUNT(film_id) AS titles FROM film; titles ----- 1000 (1 row) dvdrental=# █</pre>

SEN-209: Designing Database Systems

Lab 5

Name: Thanawin Pattanaphol

Date: 29th September 2025

```
SELECT
    AVG(F.rental_rate)
FROM film F
INNER JOIN film_category FC ON
FC.film_id = F.film_id
INNER JOIN category C ON
C.category_id = FC.category_id
WHERE C.name = 'Comedy';
```

```
dvdrental=# SELECT
    AVG(F.rental_rate)
FROM film F
INNER JOIN film_category FC ON FC.film_id = F.film_id
INNER JOIN category C ON C.category_id = FC.category_id
WHERE C.name = 'Comedy';
      avg
-----
3.1624137931034483
(1 row)
```

SEN-209: Designing Database Systems
Lab 5

Name: Thanawin Pattanaphol

Date: 29th September 2025

```
SELECT
    R.customer_id,
    COUNT(R.rental_id) AS
"Rentals"
FROM rental R
GROUP BY R.customer_id
LIMIT 30;
```

```
dvdrental=# SELECT
    R.customer_id,
    COUNT(R.rental_id) AS "Rentals"
FROM rental R
GROUP BY R.customer_id
LIMIT 30;
```

customer_id	Rentals
87	30
184	23
477	22
273	35
550	32
394	22
51	33
272	20
70	18
190	27
350	23
539	22
554	22
278	26
406	32
424	30
176	37
576	34
309	31
292	27
509	20
271	18
22	22
156	25
417	25
556	21
475	24
529	29
282	27
173	29

(30 rows)

SEN-209: Designing Database Systems

Lab 5

Name: Thanawin Pattanaphol

Date: 29th September 2025

```
SELECT
    C.first_name,
    C.last_name,
    SUM(P.amount)
FROM payment P
INNER JOIN customer C ON
C.customer_id = P.customer_id
GROUP BY C.customer_id
ORDER BY SUM(P.amount) DESC
LIMIT 5;
```

```
dvdrental=# SELECT
    C.first_name,
    C.last_name,
    SUM(P.amount)
FROM payment P
INNER JOIN customer C ON C.customer_id = P.customer_id
GROUP BY C.customer_id
ORDER BY SUM(P.amount) DESC
LIMIT 5;
 first_name | last_name |  sum
-----+-----+-----
 Eleanor    | Hunt      | 211.55
 Karl       | Seal      | 208.58
 Marion     | Snyder    | 194.61
 Rhonda    | Kennedy   | 191.62
 Clara      | Shaw      | 189.60
(5 rows)
```

```
SELECT
    C.first_name,
    C.last_name,
    COUNT(R.rental_id)
FROM rental R
INNER JOIN customer C ON
C.customer_id = R.customer_id
GROUP BY C.customer_id
ORDER BY COUNT(R.rental_id) DESC
LIMIT 5;
```

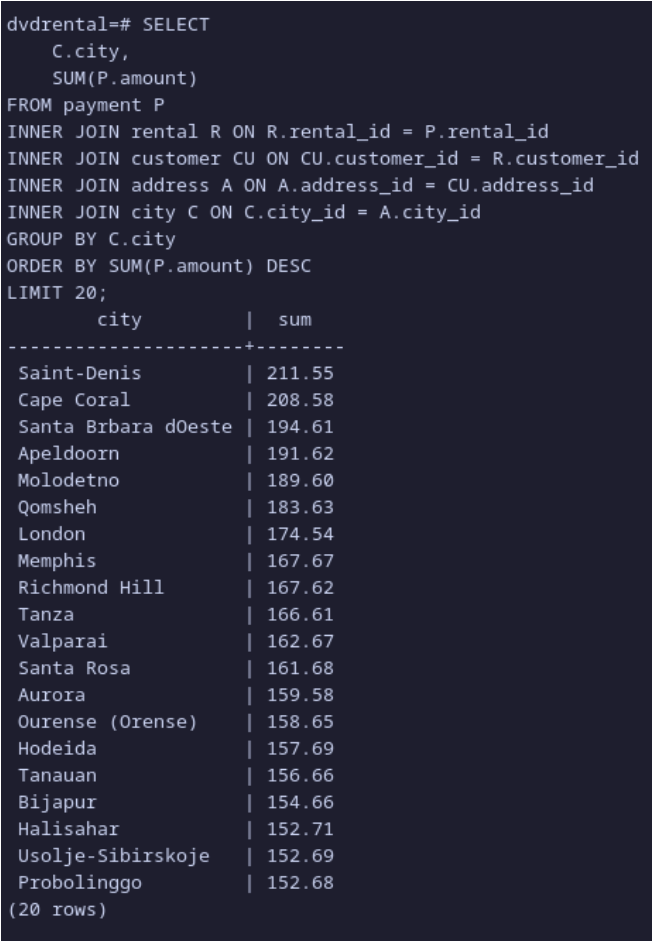
```
dvdrental=# SELECT
    C.first_name,
    C.last_name,
    COUNT(R.rental_id)
FROM rental R
INNER JOIN customer C ON C.customer_id = R.customer_id
GROUP BY C.customer_id
ORDER BY COUNT(R.rental_id) DESC
LIMIT 5;
 first_name | last_name | count
-----+-----+-----
 Eleanor    | Hunt      |    46
 Karl       | Seal      |    45
 Marcia     | Dean      |    42
 Clara      | Shaw      |    42
 Tammy      | Sanders   |    41
(5 rows)
```

SEN-209: Designing Database Systems
Lab 5

Name: Thanawin Pattanaphol

Date: 29th September 2025

Advanced Queries

SQL Query	Screenshot
<pre>SELECT C.city, SUM(P.amount) FROM payment P INNER JOIN rental R ON R.rental_id = P.rental_id INNER JOIN customer CU ON CU.customer_id = R.customer_id INNER JOIN address A ON A.address_id = CU.address_id INNER JOIN city C ON C.city_id = A.city_id GROUP BY C.city ORDER BY SUM(P.amount) DESC LIMIT 20;</pre>	 <pre>dvdrental=# SELECT C.city, SUM(P.amount) FROM payment P INNER JOIN rental R ON R.rental_id = P.rental_id INNER JOIN customer CU ON CU.customer_id = R.customer_id INNER JOIN address A ON A.address_id = CU.address_id INNER JOIN city C ON C.city_id = A.city_id GROUP BY C.city ORDER BY SUM(P.amount) DESC LIMIT 20; city sum -----+----- Saint-Denis 211.55 Cape Coral 208.58 Santa Brbara dOeste 194.61 Apeldoorn 191.62 Molodetno 189.60 Qomsheh 183.63 London 174.54 Memphis 167.67 Richmond Hill 167.62 Tanza 166.61 Valparai 162.67 Santa Rosa 161.68 Aurora 159.58 Ourense (Orense) 158.65 Hodeida 157.69 Tanauan 156.66 Bijapur 154.66 Halisahar 152.71 Usolje-Sibirskoje 152.69 Probolinggo 152.68 (20 rows)</pre>

SEN-209: Designing Database Systems

Lab 5

Name: Thanawin Pattanaphol

Date: 29th September 2025

```
SELECT
    C.name,
    COUNT(R.rental_id)
FROM film F
INNER JOIN film_category FC ON FC.film_id = F.film_id
INNER JOIN category C ON C.category_id = FC.category_id
INNER JOIN inventory I ON I.film_id = F.film_id
INNER JOIN rental R ON R.inventory_id = I.inventory_id

GROUP BY C.name
ORDER BY COUNT(R.rental_id) DESC;
```

```
dvdrental=# SELECT
    C.name,
    COUNT(R.rental_id)
FROM film F
INNER JOIN film_category FC ON FC.film_id = F.film_id
INNER JOIN category C ON C.category_id = FC.category_id
INNER JOIN inventory I ON I.film_id = F.film_id
INNER JOIN rental R ON R.inventory_id = I.inventory_id

GROUP BY C.name
ORDER BY COUNT(R.rental_id) DESC;
   name   | count
-----+-----
 Sports   |  1179
 Animation |  1166
 Action   |  1112
 Sci-Fi    |  1101
 Family   |  1096
 Drama     |  1060
 Documentary | 1050
 Foreign   |  1033
 Games     |   969
 Children  |   945
 Comedy    |   941
 New       |   940
 Classics  |   939
 Horror     |   846
 Travel    |   837
 Music     |   830
(16 rows)
```

```
SELECT DISTINCT ON (C.store_id)
    C.store_id,
    C.first_name,
    C.last_name,
    SUM(P.amount)
FROM customer C
-- INNER JOIN store S ON
S.store_id = C.store_id
INNER JOIN payment P ON
P.customer_id = C.customer_id
GROUP BY C.store_id,
C.first_name, C.last_name
ORDER BY C.store_id,
SUM(P.amount) DESC
```

```
dvdrental=# SELECT DISTINCT ON (C.store_id)
    C.store_id,
    C.first_name,
    C.last_name,
    SUM(P.amount)
FROM customer C
-- INNER JOIN store S ON S.store_id = C.store_id
INNER JOIN payment P ON P.customer_id = C.customer_id
GROUP BY C.store_id, C.first_name, C.last_name
ORDER BY C.store_id, SUM(P.amount) DESC;
 store_id | first_name | last_name | sum
-----+-----+-----+-----
        1 | Eleanor    | Hunt      | 211.55
        2 | Karl       | Seal      | 208.58
(2 rows)
```

SEN-209: Designing Database Systems

Lab 5

Name: Thanawin Pattanaphol

Date: 29th September 2025

There does not seem to be any payments in the year 2005, however, there are payments in the year 2007.

```
SELECT
    TO_CHAR(
        DATE_TRUNC('month',
payment_date),
        'YYYY-MM'
    ) AS month,
    SUM(amount)
FROM payment
WHERE EXTRACT(YEAR FROM
payment_date) = 2007
GROUP BY month;
```

```
dvdrental=# SELECT
    TO_CHAR(
        DATE_TRUNC('month', payment_date),
        'YYYY-MM'
    ) AS month,
    SUM(amount)
FROM payment
WHERE EXTRACT(YEAR FROM payment_date) = 2007
GROUP BY month;
 month | sum
-----+-----
2007-02 | 8351.84
2007-03 | 23886.56
2007-04 | 28559.46
2007-05 | 514.18
(4 rows)
```

SEN-209: Designing Database Systems

Lab 5

Name: Thanawin Pattanaphol

Date: 29th September 2025

<pre>SELECT C.first_name, C.last_name, COUNT(R.rental_id) FROM customer C INNER JOIN rental R ON R.customer_id = C.customer_id GROUP BY C.first_name, C.last_name HAVING COUNT(R.rental_id) > (SELECT AVG(rentals) FROM (SELECT customer_id, COUNT(rental_id) AS rentals FROM rental GROUP BY customer_id) AS total) ORDER BY COUNT(R.rental_id) DESC LIMIT 30;</pre>	<table><tr><th>first_name</th><th>last_name</th><th>count</th></tr><tr><td>Eleanor</td><td>Hunt</td><td>46</td></tr><tr><td>Karl</td><td>Seal</td><td>45</td></tr><tr><td>Marcia</td><td>Dean</td><td>42</td></tr><tr><td>Clara</td><td>Shaw</td><td>42</td></tr><tr><td>Tammy</td><td>Sanders</td><td>41</td></tr><tr><td>Sue</td><td>Peters</td><td>40</td></tr><tr><td>Wesley</td><td>Bull</td><td>40</td></tr><tr><td>Tim</td><td>Cary</td><td>39</td></tr><tr><td>Rhonda</td><td>Kennedy</td><td>39</td></tr><tr><td>Marion</td><td>Snyder</td><td>39</td></tr><tr><td>Curtis</td><td>Irby</td><td>38</td></tr><tr><td>Elizabeth</td><td>Brown</td><td>38</td></tr><tr><td>Tommy</td><td>Collazo</td><td>38</td></tr><tr><td>Daisy</td><td>Bates</td><td>38</td></tr><tr><td>Brandon</td><td>Huey</td><td>37</td></tr><tr><td>Marsha</td><td>Douglas</td><td>37</td></tr><tr><td>June</td><td>Carroll</td><td>37</td></tr><tr><td>Elsie</td><td>Kelley</td><td>37</td></tr><tr><td>Russell</td><td>Brinson</td><td>36</td></tr><tr><td>Alexander</td><td>Fennell</td><td>36</td></tr><tr><td>Justin</td><td>Ngo</td><td>36</td></tr><tr><td>Roger</td><td>Quintanilla</td><td>36</td></tr><tr><td>Angela</td><td>Hernandez</td><td>36</td></tr><tr><td>Margie</td><td>Wade</td><td>36</td></tr><tr><td>Michelle</td><td>Clark</td><td>35</td></tr><tr><td>Leslie</td><td>Seward</td><td>35</td></tr><tr><td>Alma</td><td>Austin</td><td>35</td></tr><tr><td>Diane</td><td>Collins</td><td>35</td></tr><tr><td>Rosemary</td><td>Schmidt</td><td>35</td></tr><tr><td>Naomi</td><td>Jennings</td><td>35</td></tr></table> <p>(30 rows)</p>	first_name	last_name	count	Eleanor	Hunt	46	Karl	Seal	45	Marcia	Dean	42	Clara	Shaw	42	Tammy	Sanders	41	Sue	Peters	40	Wesley	Bull	40	Tim	Cary	39	Rhonda	Kennedy	39	Marion	Snyder	39	Curtis	Irby	38	Elizabeth	Brown	38	Tommy	Collazo	38	Daisy	Bates	38	Brandon	Huey	37	Marsha	Douglas	37	June	Carroll	37	Elsie	Kelley	37	Russell	Brinson	36	Alexander	Fennell	36	Justin	Ngo	36	Roger	Quintanilla	36	Angela	Hernandez	36	Margie	Wade	36	Michelle	Clark	35	Leslie	Seward	35	Alma	Austin	35	Diane	Collins	35	Rosemary	Schmidt	35	Naomi	Jennings	35
first_name	last_name	count																																																																																												
Eleanor	Hunt	46																																																																																												
Karl	Seal	45																																																																																												
Marcia	Dean	42																																																																																												
Clara	Shaw	42																																																																																												
Tammy	Sanders	41																																																																																												
Sue	Peters	40																																																																																												
Wesley	Bull	40																																																																																												
Tim	Cary	39																																																																																												
Rhonda	Kennedy	39																																																																																												
Marion	Snyder	39																																																																																												
Curtis	Irby	38																																																																																												
Elizabeth	Brown	38																																																																																												
Tommy	Collazo	38																																																																																												
Daisy	Bates	38																																																																																												
Brandon	Huey	37																																																																																												
Marsha	Douglas	37																																																																																												
June	Carroll	37																																																																																												
Elsie	Kelley	37																																																																																												
Russell	Brinson	36																																																																																												
Alexander	Fennell	36																																																																																												
Justin	Ngo	36																																																																																												
Roger	Quintanilla	36																																																																																												
Angela	Hernandez	36																																																																																												
Margie	Wade	36																																																																																												
Michelle	Clark	35																																																																																												
Leslie	Seward	35																																																																																												
Alma	Austin	35																																																																																												
Diane	Collins	35																																																																																												
Rosemary	Schmidt	35																																																																																												
Naomi	Jennings	35																																																																																												

Reflection

The easiest questions were certainly mostly located in the basic queries since it does not require joining tables, sub-queries or other aggregate functions. The more difficult ones are mostly ones in the advanced queries; these require a substantial amount of table joins, aggregate functions and careful consideration of which function or sub-query is appropriate in each question.