```sql
-----------------------------------------
--String Functions - Letter Case
-----------------------------------------

SELECT title,
      LOWER(title) lo_title,
      UPPER(title) up_title,
      INITCAP(title) ic_title
FROM film;

SELECT  *
FROM film
WHERE LOWER(title) LIKE '%trip%'

-----------------------------------------
--String Functions - Character Processing
-----------------------------------------

SELECT first_name, last_name,
      CONCAT(first_name, ' ', last_name) as f1,
      CONCAT_WS(',', first_name, last_name) as f2,
      LEFT(first_name, 1) as f3,
      RIGHT(first_name, 1) as f4,
      LPAD(phone_number, 14, '00') as f6,
      RPAD(phone_number, 14, '00') as f7,
FROM employees;

SELECT first_name, last_name,
      LENGTH(last_name) as f5,
      REPLACE(first_name, 'e', ' * ') as f8,
      SPLIT_PART(hire_date::varchar, '-', 2) as f9,
      SUBSTRING(last_name, 2, 3) as f10,
      POSITION('a' in first_name) as f11,
      REVERSE(first_name) as f12
FROM employees;

-----------------------------------------
--Math Functions
-----------------------------------------

SELECT ROUND(14.45),
       CEIL(14.45),
       FLOOR(14.45),
       ABS(-5.78),
       POWER(4,3),
       SIGN(-5),
       TRUNC(4836.98);

SELECT payment_id, amount,
      (amount * 0.45) as percent_amount,
      ROUND(amount * 0.45) as f_round,
      CEIL(amount * 0.45) as f_ceil,
      FLOOR(amount * 0.45) as f_floor,
      TRUNC(amount * 0.45) as f_trunc,
      MOD(amount, 5) as f_mod
FROM payment
LIMIT 10;
```

```sql
-----------------------------------------
--Math Functions-Random
-----------------------------------------

SELECT random();

SELECT random() * 100 + 1 as ran_num;

SELECT floor(random() * 100 + 1)::int as ran_num;

SELECT floor(random() * (high - low + 1) + low)::int as ran_num;

SELECT floor(random() * (200 - 100 + 1) + 100)::int as ran_num
FROM generate_series(1, 10);

-----------------------------------------
--Date Functions
-----------------------------------------

SELECT CURRENT_DATE,
       CURRENT_TIME,
       LOCALTIME,
       NOW(),
       TIMEOFDAY();

SELECT first_name,
       last_name,
       hire_date,
       AGE(hire_date) as age_of_hire
FROM employees;

-----------------------------------------
--Date Functions - DATE_PART
-----------------------------------------

SELECT CURRENT_DATE,
      DATE_PART('century', CURRENT_DATE) as century_,
      DATE_PART('quarter', CURRENT_DATE) as quarter_,
      DATE_PART('decade', CURRENT_DATE) as decade_,
      DATE_PART('year', CURRENT_DATE) as year_,
      DATE_PART('month', CURRENT_DATE) as month_,
      DATE_PART('day', CURRENT_DATE) as day_,
      DATE_PART('hour', CURRENT_DATE) as hour_,
      DATE_PART('minute', CURRENT_DATE) as minute_,
      DATE_PART('dow', CURRENT_DATE) as dow_,
      DATE_PART('doy', CURRENT_DATE) as doy_,
      DATE_PART('timezone', CURRENT_TIME) as timezone_;
```

```sql
-------------------------------------------
--Date Functions - EXTRACT
-------------------------------------------

SELECT CURRENT_DATE,
      EXTRACT(CENTURY FROM CURRENT_DATE) as century_,
      EXTRACT(QUARTER FROM CURRENT_DATE) as quarter_,
      EXTRACT(DECADE FROM CURRENT_DATE) as decade_,
      EXTRACT(YEAR FROM CURRENT_DATE) as year_,
      EXTRACT(MONTH FROM CURRENT_DATE) as month_,
      EXTRACT(DAY FROM CURRENT_DATE) as day_,
      EXTRACT(HOUR FROM CURRENT_TIME) as hour_,
      EXTRACT(MINUTE FROM CURRENT_TIME) as minute_,
      EXTRACT(DOW FROM CURRENT_DATE) as dow_,
      EXTRACT(DOY FROM CURRENT_DATE) as doy_,
      EXTRACT(TIMEZONE FROM CURRENT_TIME) as timezone_;

-------------------------------------------
--Date Functions - DATE_TRUNC
-------------------------------------------

SELECT
       DATE_TRUNC('quarter', TIMESTAMP '2023-04-17 06:12:38') as quarter_,
       DATE_TRUNC('year',    TIMESTAMP '2023-04-17 06:12:38') as year_,
       DATE_TRUNC('month',   TIMESTAMP '2023-04-17 06:12:38') as month_,
       DATE_TRUNC('day',     TIMESTAMP '2023-04-17 06:12:38') as day_,
       DATE_TRUNC('hour',    TIMESTAMP '2023-04-17 06:12:38') as hour_,
       DATE_TRUNC('minute',  TIMESTAMP '2023-04-17 06:12:38') as minute_,
       DATE_TRUNC('second',  TIMESTAMP '2023-04-17 06:12:38') as second_

-------------------------------------------
--Conversion Functions - TO_DATE
-------------------------------------------

SELECT TO_DATE('20230405', 'YYYYMMDD') TD1,
       TO_DATE('2023 APRIL 05', 'YYYY MONTH DD') TD2,
       TO_DATE('2023 april 05', 'YYYY month DD') TD3,
       TO_DATE('2023 APR 05', 'YYYY MON DD') TD4,
       TO_DATE('2023 240', 'YYYY DDD') TD5,
       TO_DATE('February 08, 2023', 'Month DD, YYYY') TD6;

-------------------------------------------
--Conversion Functions - TO_TIMESTAMP
-------------------------------------------

SELECT TO_TIMESTAMP('2021-05-30 08:40:30', 'YYYY-MM-DD HH:MI:SS') TT1,
       TO_TIMESTAMP('05.30.2021 20:40:30', 'MM.DD.YYYY HH24:MI:SS') TT2,
       TO_TIMESTAMP('2022/25/08 08:40', 'YYYY/DD/MM HH:MI') TT3,
       TO_TIMESTAMP('11 30 99 12:40', 'MM DD YY HH:MI') TT4,
       TO_TIMESTAMP('09 07 19 10:35', 'MM DD YY HH:MI') TT5,
       TO_TIMESTAMP('2022 OCT 15 07:21:11', 'YYYY MON DD HH:MI:SS') TT6;
```

```sql
-------------------------------------------
--Conversion Functions - TO_NUMBER
-------------------------------------------

SELECT TO_NUMBER('1210.73', '9999.99') TN1,
       TO_NUMBER('1,210.73', '9G999.99') TN2,
       TO_NUMBER('$1,210.73', 'L9G999.99') TN3,
       TO_NUMBER('$1,210.73', 'L9G999') TN4,
       TO_NUMBER('-12.345,6', '99G999D9S') TN5,
       TO_NUMBER('$1.234.567,89', 'L9G999g999,99') TN6,
       TO_NUMBER('00005469', '9999999999') TN7,
       '00005469'::integer TN8,
       CAST('00005469' as integer) TN9;

-------------------------------------------
--Conversion Functions - TO_CHAR
-------------------------------------------

SELECT payment_id, payment_date, amount,
       TO_CHAR(payment_date, 'HH24:MI:SS')            as TC1,
       TO_CHAR(payment_date, 'MON-DD-YYYY HH12:MI PM') as TC2,
       TO_CHAR(payment_date, 'DD.MM.YYYY HH24:MI')    as TC3,
       TO_CHAR(payment_date, 'MON-DAY-YYYY HH12:MI')  as TC4,
       TO_CHAR(payment_date, 'Month DD, YYYY')        as TC5,
       TO_CHAR(payment_date, 'YYYYMMDD')              as TC6,
       TO_CHAR(amount,       '99D99')                 as TC7
FROM payment;

-------------------------------------------
--CAST Function
-------------------------------------------

SELECT
    CAST ('100' as INTEGER) as cast1,
    CAST ('2021-01-01' as DATE) as cast2,
    CAST ('15-OCT-2022' as DATE) as cast3,
    CAST ('10.25' as DOUBLE PRECISION) as cast4,
    CAST ('true' as BOOLEAN) as cast5,
    CAST ('false' as BOOLEAN) as cast6,
    CAST ('T' as BOOLEAN) as cast7,
    CAST ('F' as BOOLEAN) as cast8;

SELECT
    '100'::INTEGER  as cast1,
    '01-OCT-2015'::DATE as cast2,
    598::VARCHAR as cast3,
    '2019-06-15 14:30:20'::timestamp as cast4,
    '15 minute'::interval as cast5,
    '2 hour'::interval as cast6,
    '1 day'::interval as cast7,
    '2 week'::interval as cast8,
    '3 month'::interval as cast9;
```

```sql
SELECT
    CAST ('2 year 5 months 3 days' AS INTERVAL),
    CAST (2800 AS MONEY),
    CAST (CURRENT_DATE AS TEXT);

SELECT
    date_value_str,
    CAST (date_value_str AS DATE)
FROM date_demo;

-------------------------------------------
--Arithmetic Operations with Dates
-------------------------------------------

SELECT current_date, current_time,
       current_date + 10 as F1,
       current_date - 5 as F2,
       current_date - TO_DATE('01012022', 'DDMMYYYY') as F3,
       current_time + INTERVAL '2 hour' as F4,
       NOW() - INTERVAL '1 year 2 months 3 hours 20 minutes' as F5;

-------------------------------------------
--COLAESCE Function
-------------------------------------------

SELECT
    COALESCE(1, 2, 3) C1,
    COALESCE(null, 2, 3) C2,
    COALESCE(null, null, 3) C2;

SELECT phone_number ,
       COALESCE(phone_number, 'No phone number')
FROM employees;

-------------------------------------------
--NULLIF Function
-------------------------------------------

SELECT
    NULLIF(1, 1) N1,
    NULLIF(1, 2) N2,
    NULLIF('A', 'B') N3;

SELECT b.*,
       NULLIF(current_year, previous_year) as budget
FROM budgets b;
```

```sql
-----------------------------------------
--CASE Expression
-----------------------------------------

SELECT title, length,
    CASE
        WHEN length >= 0   AND length <= 50 THEN 'Short length'
        WHEN length >= 51  AND length <= 120 THEN 'Medium length'
        WHEN length > 120 THEN 'Long length'
    END duration
FROM film
ORDER BY title;

SELECT title, rating,
    CASE rating
        WHEN 'G' THEN 'General Audiences'
        WHEN 'PG' THEN 'Parental Guidance Suggested'
        WHEN 'PG-13' THEN 'Parents Strongly Cautioned'
        WHEN 'R' THEN 'Restricted'
        WHEN 'NC-17' THEN 'Adults Only'
    END rating_description
FROM film
ORDER BY title;

SELECT first_name, last_name, job_id,
    CASE
        WHEN job_id in (2,7,10,14,15,19) THEN 'Manager Positions'
        WHEN job_id in (13,17,18) THEN 'Clerk Positions'
        ELSE 'Other Positions'
    END position_type
FROM employees;

-----------------------------------------
--Nested Functions
-----------------------------------------

SELECT first_name, last_name,
    LENGTH(CONCAT(first_name, last_name)) as length_name,
    CONCAT(SUBSTRING(first_name, 1, 2), '.', SUBSTRING(last_name, 1, 2), '.') as
name
FROM employees

SELECT salesman_id,
  COALESCE(
          CAST(
                NULLIF(current_year, previous_year)
                as Varchar),
          'Same as last year') as budget
FROM budgets
WHERE current_year IS NOT NULL;
```

```sql
-----------------------------------------------------
--EXERCISE ANSWERS
-----------------------------------------------------

-------------------------------------------
--String Functions - Letter Case
-------------------------------------------

select
    upper(title) as title_new,
    lower(description) as description_new
from film
where
    lower(description) like '%drama%'
    and lower(description) like '%australia%'

-------------------------------------------
--String Functions - Character Processing
-------------------------------------------

--Exercise-1:

SELECT title, description FROM film
WHERE description like '%Hunter%'

SELECT title, description FROM film
WHERE initcap(description) like '%Hunter%'

SELECT title, description FROM film
WHERE position('Hunter' in description) > 0


--Exercise-2:

SELECT first_name, last_name,
       CONCAT(first_name, ' ', last_name, ' ', email) as f1,
       CONCAT_WS(' ', first_name, last_name, email) as f2,
       first_name || ' ' || last_name || ' ' || email as f3
FROM employees;

-------------------------------------------
--Date Functions
-------------------------------------------

SELECT
    'Quarter is: ' || DATE_PART('quarter', CURRENT_DATE) || ', '
    'Year is: '    || DATE_PART('year', CURRENT_DATE)    || ', '
    'Month is: '   || DATE_PART('month', CURRENT_DATE)   || ', '
    'Doy is: '     || DATE_PART('doy', CURRENT_DATE)     || '.'

SELECT
    'Quarter is: ' || EXTRACT(quarter from CURRENT_DATE) || ', '
    'Year is: '    || EXTRACT(year from CURRENT_DATE)    || ', '
    'Month is: '   || EXTRACT(month from CURRENT_DATE)   || ', '
    'Doy is: '     || EXTRACT(doy from CURRENT_DATE)     || '.'
```

```sql
-----------------------------------------
--Conversion Functions - TO_CHAR
-----------------------------------------

SELECT customer_id
       || ' paid: ' ||
       TO_CHAR(amount,'$99D99')
       || ' at ' ||
       TO_CHAR(payment_date, 'HH24:MI:SS')
       || ' on ' ||
       TO_CHAR(payment_date, 'Mon-DD-YYYY') as p_info
FROM payment
ORDER BY rental_id
LIMIT 5;

-----------------------------------------
--COLAESCE Function
-----------------------------------------

SELECT
    brand, price, discount,
    (price - COALESCE(discount,0)) as net_price
FROM cars;

-----------------------------------------
--NULLIF Function
-----------------------------------------

SELECT b.*,
       COALESCE(current_year, previous_year) as budget1,
       COALESCE(NULLIF(current_year,NULL), previous_year) as budget2
FROM budgets b;

-----------------------------------------
--CASE Expression
-----------------------------------------

select first_name, last_name, hire_date, salary,
  case
    when date_part('year', hire_date) between 2018 and 2020 then salary * 1.25
    when date_part('year', hire_date) < 2018 then salary * 1.35
    when date_part('year', hire_date) > 2020 then salary * 1.15
  end as new_salary
from employees
```