

READ

```
SELECT customernumber, customername,  
contactlastname || ', ' || contactfirstname AS "Fullname"  
FROM customers  
WHERE LOWER(city) = 'paris';
```

```
SELECT DISTINCT customernumber  
FROM payments  
WHERE amount IS NOT NULL; /* amount > 0 */  
  
WHERE amount < 30000 OR amount > 65000 /* using NOT BETWEEN 30000 AND 65000 */
```

CREATE

```
INSERT INTO employees  
(employeenumber, lastname, firstname, extension, email, officecode, reportsto, jobtitle)  
VALUES  
(1704, 'Dugar', 'Divyanshu', 'x905', 'ddugar@myseneca.ca', '4', 1088, 'Cashier');  
  
INSERT ALL  
INTO employees VALUES (1704, 'Dugar', 'Divyanshu', 'x905', 'ddugar@myseneca.ca', '4', 1088, 'Head Cashier')  
INTO employees VALUES (1706, 'Kaur', 'Rehatpreet', 'x647', 'rpkaaur4@myseneca.ca', '5', 1088, 'Cashier')  
SELECT * FROM dual;
```

UPDATE

```
UPDATE employee2 SET  
firstname = 'Divyanshu',  
lastname = 'Dugar'  
WHERE employeenumber = 1002;  
  
UPDATE employees SET  
jobtitle = 'Head Cashier'  
WHERE  
employeenumber = 1704;  
  
UPDATE employee2 SET  
username = LOWER(CONCAT(SUBSTR(firstname,1,1),lastname));
```

DELETE

```
DELETE FROM employees  
WHERE employeenumber = 1704;  
  
DELETE FROM employees  
WHERE employeenumber IN(1704,1706);  
  
DELETE FROM employee2  
WHERE officecode = 4;
```

CREATING TABLES

```
CREATE TABLE movies (  
mid INT PRIMARY KEY,  
title VARCHAR(35) NOT NULL,  
releaseYear INT NOT NULL,  
director INT NOT NULL,  
score DECIMAL(3,2) CHECK (score BETWEEN 0 AND 5)  
);  
  
CREATE TABLE castings (  
movieid INT,  
actorid INT,  
PRIMARY KEY (movieid, actorid),  
CONSTRAINT movie_id FOREIGN KEY (movieid) REFERENCES movies(mid),  
CONSTRAINT actor_id FOREIGN KEY (actorid) REFERENCES actors(aid)  
);  
  
CREATE TABLE pgm(  
pgm_id VARCHAR(3) PRIMARY KEY,  
pgm_name VARCHAR(40) NOT NULL UNIQUE,  
no_semesters NUMBER(1)DEFAULT 6 NOT NULL,  
CONSTRAINT chk_pgmid CHECK (pgm_id IN ('CPA', 'CPD')),  
CONSTRAINT chk_numSem CHECK (no_semesters >= 0)  
);  
  
CREATE TABLE employee2 AS  
(SELECT * FROM employees);
```

ALTER TABLE

```
ALTER TABLE table_name  
MODIFY column_name type constraint;  
  
ALTER TABLE table_name  
DROP ( column_name_1,  
column_name_2  
);  
  
ALTER TABLE movies  
ADD CONSTRAINT director_fk  
FOREIGN KEY (director)  
REFERENCES directors(directorid);  
  
ALTER TABLE movies  
ADD CONSTRAINT title_unq  
UNIQUE(title);  
  
ALTER TABLE employee2  
ADD username VARCHAR(20);
```

INNER JOIN – DISPLAYS ONLY THE COLUMNS WHO SATISFY THE ON CONDITION || LEFT JOIN – DISPLAYS ALL THE COLUMNS

INNER JOINS/JOINS

```
SELECT e.employeenumber, e.firstname, e.lastname, c.phone, c.postalcode  
FROM EMPLOYEES e  
INNER JOIN CUSTOMERS c  
ON c.salesrepemployeenumber = e.employeenumber  
WHERE LOWER(country) = 'france';  
  
SELECT c.*, p.*  
FROM CUSTOMERS c  
INNER JOIN PAYMENTS p  
ON p.customernumber = c.customernumber  
WHERE LOWER(country) = 'canada'  
ORDER BY c.customernumber;
```

CREATING/REPLACING VIEWS BY INNER JOINING MULTIPLE TABLES

```
CREATE VIEW vwCustomerOrder AS
SELECT c.customernumber, o.ordernumber, o.orderdate, p.productname, od.quantityordered, od.priceeach
FROM customers c
JOIN orders o ON o.customernumber = c.customernumber
JOIN orderdetails od ON od.ordernumber = o.ordernumber
JOIN products p ON p.productcode = od.productcode;
```

```
CREATE OR REPLACE VIEW vwCustomerOrder AS
SELECT c.customernumber, o.ordernumber, o.orderdate, p.productname,
od.quantityordered, od.priceeach, od.orderlinenumber
FROM customers c
JOIN orders o ON o.customernumber = c.customernumber
JOIN orderdetails od ON od.ordernumber = o.ordernumber
JOIN products p ON p.productcode = od.productcode;
```

VIEWING A VIEW -> `SELECT * FROM vwcustomerorder;`

CREATING/REPLACING VIEWS BY LEFT JOINING 2 TABLES

```
CREATE OR REPLACE VIEW vwEmployeeManager AS
SELECT e.*, m.firstname AS ManagerFirstName, m.lastname AS ManagerLastName
FROM employees e
LEFT JOIN employees m
ON e.reportsto = m.employeenumber ;
```

USING TO_CHAR FUNCTION

```
SELECT c.customernumber, c.customername, TO_CHAR(p.paymentdate, 'MON DD,YYYY'), p.amount
FROM CUSTOMERS c
INNER JOIN PAYMENTS p
ON p.customernumber = c.customernumber
WHERE LOWER(country) = 'canada'
ORDER BY c.customernumber;
```

DROPPING TABLES AND VIEWS

```
DROP VIEW vwcustomerorder; DROP TABLE employee2;
```

Create a query that lists the teams that do not yet have any players linked to them:

```
SELECT teamID, teamName
FROM teams LEFT JOIN players
ON TEAMS.TEAMID=players.teamid
WHERE playerID IS NULL;
```

```
UPDATE staff SET
active = 0
WHERE empid = 1005;
```

```
-- FIRING --
UPDATE customer SET
empid = NULL
WHERE custid = 1050;
```

```
DELETE FROM staff
WHERE empid = 1005;
```

```
SELECT *
FROM orders
FETCH NEXT 10 ROWS ONLY;
```

```
SELECT * FROM dual; SELECT 2 * 7 AS number FROM dual;
```