```
#ALTER
ALTER TABLE table_name
ADD column_name datatype;
ALTER TABLE table_name
DROP COLUMN column_name;
My SQL / Oracle (prior version 10G):
ALTER TABLE table_name
MODIFY COLUMN column_name datatype;
Oracle 10G and later:
ALTER TABLE table_name
MODIFY column_name datatype;
```

```
CREATE TABLE employee (
  emp_id INT ,
  first_name VARCHAR(40),
  super_id INT,
  branch_id INT,
  CONSTRAINT pk_employee PRIMARY KEY (emp_id),
  CONSTRAINT fk_employee_employee FOREIGN KEY (super_id)
   REFERENCES employee(emp_id) ON DELETE SET NULL
);
```

```
CREATE TABLE branch(
  branch_id INT,
  branch_name VARCHAR(40),
  mgr_id INT,
  mgr_start_date DATE,

  CONSTRAINT pk_branch PRIMARY KEY (branch_id),
  CONSTRAINT fk_branch_employee FOREIGN KEY (mgr_id)
    REFERENCES employee(emp_id) ON DELETE SET NULL);
```

```
ALTER table employee
ADD CONSTRAINT fk_employee_branch FOREIGN KEY (branch_id)
REFERENCES branch(branch_id)
ON DELETE SET NULL ;
```

```
INSERT INTO Teacher (Tcode, TFName, TLName) VALUES ('T04',      'อภิศักดิ์'       ,
'พัฒนจักร');
INSERT INTO Borrow VALUES(1, '623021039-4'         , 'B01',
TO_DATE('24/2/2015','DD/MM/YYYY'), TO_DATE('25/2/2015','DD/MM/YYYY'));
```

```
CREATE VIEW  LAB6_BookBorrow AS
SELECT BOOKID, BOOKNAME, COUNT(*) - 1 Amount
FROM(
  SELECT BOOKID, BOOKNAME
  FROM BOOK
  UNION ALL
  SELECT BORROW.BOOKID, BOOK.BOOKNAME
  FROM BORROW
    INNER JOIN BOOK  on BOOK.BOOKID = BORROW.BOOKID
  )
GROUP BY BOOKID, BOOKNAME
ORDER BY BOOKID;
```

```
select 'i like' || type_column || ' with'
ect....
```

```
#Other important keywords
LEFT OUTER JOIN, MIN, MAX, AVG, SUM
```

```
WITH a AS
   ( SELECT name, birthdate, YEAR(birthdate) AS birthyear
     FROM persons
     WHERE birthdate >= '1950-01-01' AND birthdate <
'1960-01-01'
   )
, b AS
   ( SELECT birthyear, COUNT(*) AS cnt
     FROM a
     GROUP BY birthyear
   )
SELECT a.name, a.birthdate, b.cnt AS number_of_births
FROM a JOIN b
  ON a.birthyear = b.birthyear
WHERE MONTH(a.birthdate) = 7 ;
```

```
SELECT DISTINCT column1, column2, ...
FROM table_name;

ELECT * FROM Customers
ORDER BY Country ASC, CustomerName DESC;

WHERE column_name IS NULL;
WHERE column_name IS NOT NULL;

UPDATE table_name
SET column1 = value1, column2 = value2,
...
WHERE condition;
```

```
DELETE FROM Customers
WHERE CustomerName='Alfreds Futterkiste';
```

```
SELECT column_name(s)
FROM table_name
WHERE ROWNUM <= number;
```

```
SELECT MIN(column_name)  #MAX
FROM table_name
WHERE condition;
```

```
SELECT column1, column2, ...
FROM table_name
WHERE columnN LIKE pattern;
```

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1, value2,
...);
```

```
SELECT * FROM Customers
WHERE Country IN (SELECT Country FROM Su
ppliers);
```

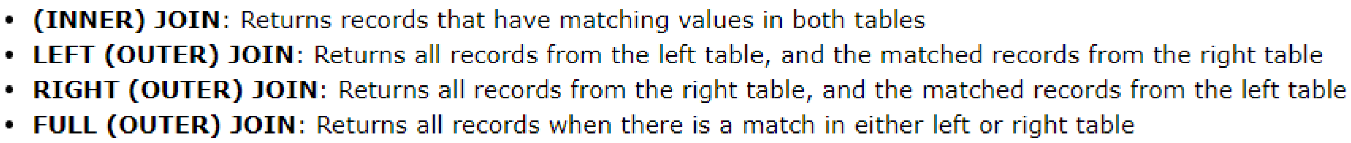
```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND val
ue2;
```

```
SELECT OrderID, Quantity,
CASE
    WHEN Quantity > 30 THEN 'The
quantity is greater than 30'
    WHEN Quantity = 30 THEN 'The
quantity is 30'
    ELSE 'The quantity is under 30'
END AS QuantityText
FROM OrderDetails;
```

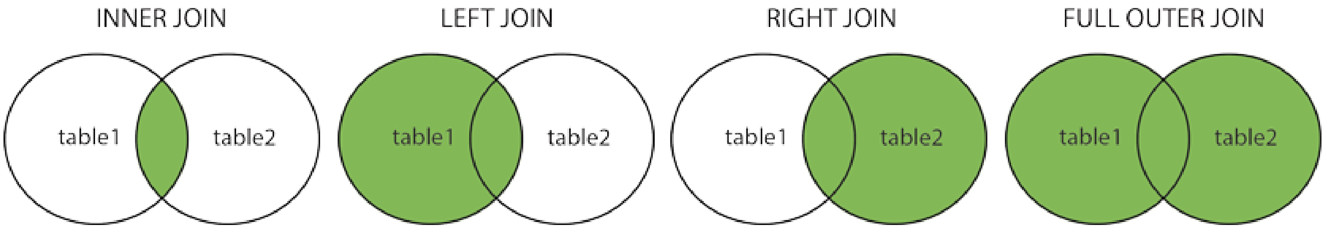
```
SELECT Employees.LastName, COUNT(Orders.
OrderID) AS NumberOfOrders
FROM (Orders
INNER JOIN Employees ON Orders.EmployeeI
D = Employees.EmployeeID)
GROUP BY LastName
HAVING COUNT(Orders.OrderID) > 10;
```

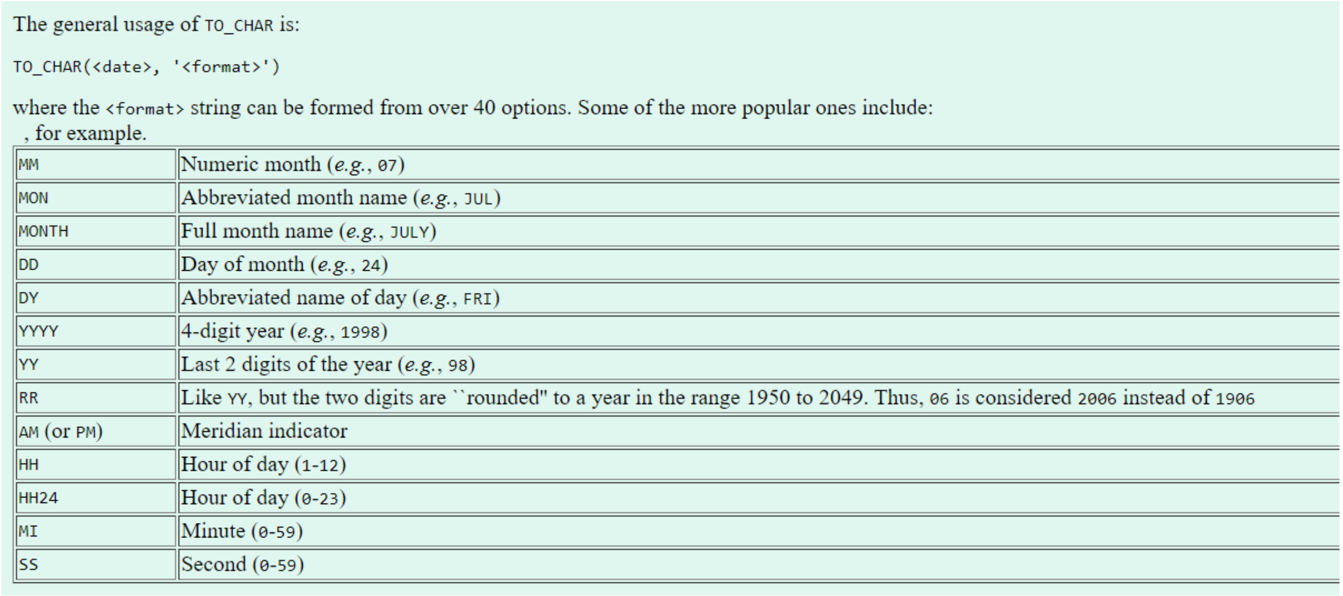
```
ALTER TABLE table_name
ADD column_name datatype;
```

```
ALTER TABLE table_name
DROP COLUMN column_name;
```



- **(INNER) JOIN**: Returns records that have matching values in both tables
- **LEFT (OUTER) JOIN**: Returns all records from the left table, and the matched records from the right table
- **RIGHT (OUTER) JOIN**: Returns all records from the right table, and the matched records from the left table
- **FULL (OUTER) JOIN**: Returns all records when there is a match in either left or right table



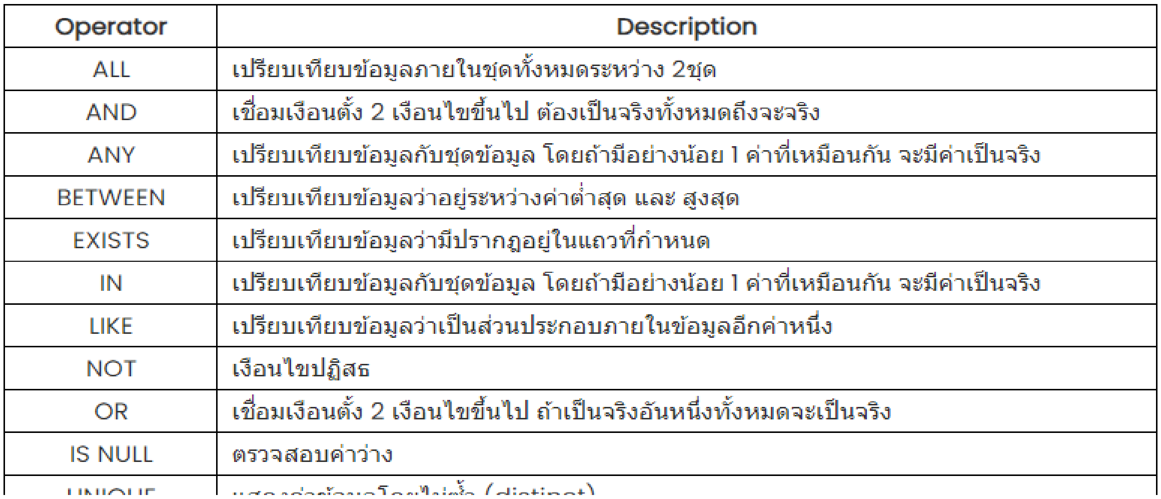


The general usage of TO_CHAR is:

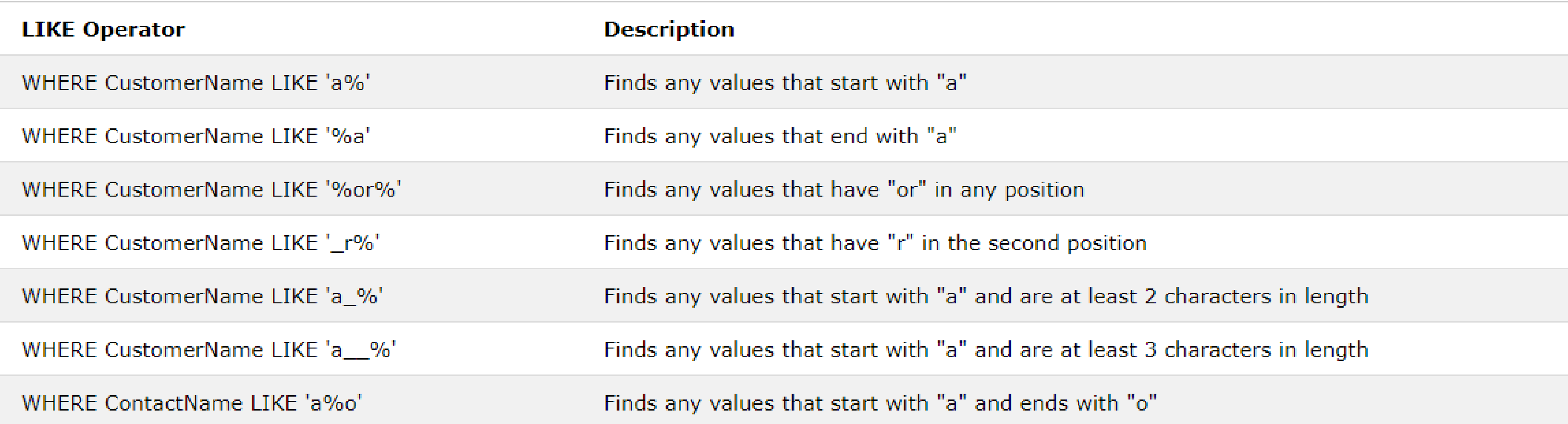
```
TO_CHAR(<date>, '<format>')
```

where the `<format>` string can be formed from over 40 options. Some of the more popular ones include:
, for example.

MM	Numeric month (*e.g.*, 07)
MON	Abbreviated month name (*e.g.*, JUL)
MONTH	Full month name (*e.g.*, JULY)
DD	Day of month (*e.g.*, 24)
DY	Abbreviated name of day (*e.g.*, FRI)
YYYY	4-digit year (*e.g.*, 1998)
YY	Last 2 digits of the year (*e.g.*, 98)
RR	Like YY, but the two digits are ``rounded'' to a year in the range 1950 to 2049. Thus, 06 is considered 2006 instead of 1906
AM (or PM)	Meridian indicator
HH	Hour of day (1-12)
HH24	Hour of day (0-23)
MI	Minute (0-59)
SS	Second (0-59)



Operator	Description
ALL	เปรียบเทียบข้อมูลภายในชุดทั้งหมดระหว่าง 2ชุด
AND	เชื่อมเงือนตั้ง 2 เงือนไขขึ้นไป ต้องเป็นจริงทั้งหมดถึงจะจริง
ANY	เปรียบเทียบข้อมูลกับชุดข้อมูล โดยถ้ามีอย่างน้อย 1 ค่าที่เหมือนกัน จะมีค่าเป็นจริง
BETWEEN	เปรียบเทียบข้อมูลว่าอยู่ระหว่างค่าต่ำสุด และ สูงสุด
EXISTS	เปรียบเทียบข้อมูลว่ามีปรากฎอยู่ในแถวที่กำหนด
IN	เปรียบเทียบข้อมูลกับชุดข้อมูล โดยถ้ามีอย่างน้อย 1 ค่าที่เหมือนกัน จะมีค่าเป็นจริง
LIKE	เปรียบเทียบข้อมูลว่าเป็นส่วนประกอบภายในข้อมูลอีกค่าหนึ่ง
NOT	เงือนไขปฏิสธ
OR	เชื่อมเงือนตั้ง 2 เงือนไขขึ้นไป ถ้าเป็นจริงอันหนึ่งทั้งหมดจะเป็นจริง
IS NULL	ตรวจสอบค่าว่าง
UNIQUE	แสดงค่าข้อมูลโดยไม่ซ้ำ (distinct)



LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that start with "a"
WHERE CustomerName LIKE '%a'	Finds any values that end with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%'	Finds any values that start with "a" and are at least 2 characters in length
WHERE CustomerName LIKE 'a__%'	Finds any values that start with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that start with "a" and ends with "o"

```
SELECT SupplierName

FROM Suppliers

WHERE EXISTS (SELECT ProductName FROM Products WHERE

Products.SupplierID = Suppliers.supplierID AND Price < 20);
```

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator ANY | ALL
(SELECT column_name FROM table_name WHER
E condition);
```

Note: The operator must be a standard comparison operator (=, <>, !=, >, >=, <, or <=).

```
CREATE USER ucs2

IDENTIFIED BY p123

DEFAULT TABLESPACE users

TEMPORARY TABLESPACE temp

QUOTA UNLIMITED ON users;

//ให สิทธิ์

GRANT resource, connect To ucs2;

GRANT <prib/rol> TO <myUser>
```