SQL

```
Select column from table
select Mert AS result
Select * from films
```

```
Select name, year, id
from Films;
```

```
Select distinct country from films
Select COUNT(*) from people
select COUNT(distinct birthday) from people
```

```
Select * from films where year = 2016
select count(*) from films where year < 2000
                                  language = 'french'
```

```
select * from films
where release_year > 2000
and language = 'Spanish';
```

```
select title, year
from films
where (year > 1989 and year < 2000)
and (language = 'French' or language = 'Spanish')
```

```
select title from films
where year between 1990 and 2000 ;
```

```
select name from kids
where age in (2,4,6,8);
```

```
select name from people
where deathdate is NULL
```

```
select name from people
where name LIKE 'B%'  # % zero or more
                '_r%'  # '_' char
           NOT LIKE 'A%'
```

```
select SUM(duration) from films
AVG(), MIN(), MAX()    # aggregate functions
```

```
select title, (gross - budget) as net_profit
from films;
```

```
Select title from films
ORDER BY release_year DESC  # descending
```

```
- - -     - - -
ORDER BY year, gross  # Multiple sort by
```

```
Select sex, count(*)
from employees
GROUP BY sex;
                    #perform operations
                          by group
```

```
Select release_year, count(*) from films
group by release_year;
```

```
select country, release_year, min(gross)
from films
group by country, release_year
order by country, release_year;
```

```
Select release_year
from films
Group by release_year
having count(title)>200
order by release_year;
limit 5 ; # show only 5
```

```
Select  title, imdb_score
from films
JOIN reviews
ON  films.id = reviews.film_id
where title ='To Kill a Mockingbird'
```

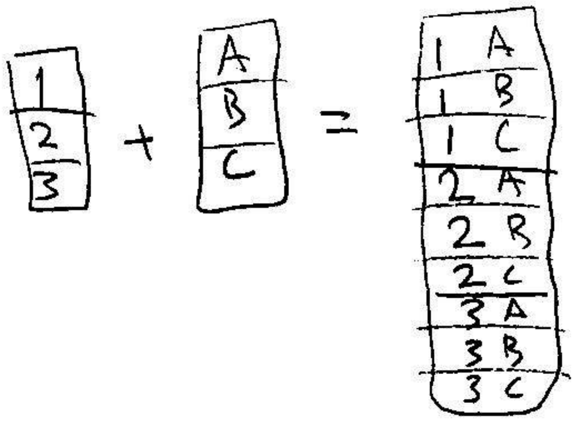
```
select ab.c as a
       de.f as d
from ab
inner join de
using(c)
```

```
SELECT name, continent, indep_year,
      CASE WHEN indep_year<1900
           THEN 'Before 1900'
           WHEN indep_year<=1930
           THEN 'between 1900 and 1930'
           ELSE 'after 1930' END
           AS indep_year_group
FROM states
ORDER BY indep_year_group
```

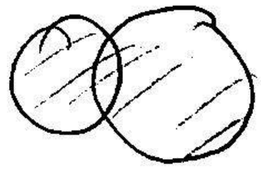
SQL

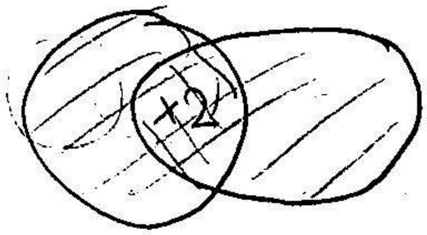
INNER JOIN
LEFT JOIN
RIGHT JOIN
FULL JOIN

CROSS JOIN

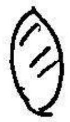


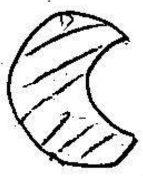
Page 8


Union



Union ALL


intersect


Except

```
select prime_minister as leader, country
from prime_ministers
UNION
select monarch, country
from monarchs
ORDER BY country;
```

```
select name
from states
where indep_year<1800;
```

SEMI-JOIN

+

```
select president, country, continent
from presidents;
```

=

```
select president, country, continent
from presidents
where country IN
        (select name
         from states
         where indep_year<1800);
```

```
select president, country, continent
from presidents
where continent LIKE '% America'
        AND country NOT IN
            (select name
             from states
             where indep_year<1800);
```

ANTI-JOIN

Asian Countries Below Average Fertility

```
Select name , fert_rate
from states
where continent ='Asia'
  AND fert_rate <
      (select avg(fert_rate)
      from states);
```

```
Select DISTINCT monarchs.continent,
subquery.max_perc
from monarchs,
    (Select continent, MAX(women_parli_perc) as max_perc
     From states
     Group by continent) AS subquery
where monarchs.continent = subquery.continent
ORDER By continent;
```

Continent	max_perc
Asia	2,4
Europe	39,6

Cross Join

```
Select  table1.id  as id1,
        table2.id  as id2
from table1
CROSS JOIN table2;
```

SQL

CASE Statement

```
CASE  WHEN  x=1  THEN 'a'
      WHEN  x=2  THEN 'b'
      Else    END  outcome
```

```
Select  date,
   Case  when   home_goal > away_goal then
                'HomeWin'
         when home_goal < away_goal then
              'Home Loss'
         else 'Tie'  end  as  outcome
```

SUBQUERY

Can be placed in any part of the query.

Select, from, where, group by

```
select  home_goal
from  Match
Where home_goal >(Select AVG(home_goal)
FROM Match);
```

Nested Subquery

Correlated Subquery

> uses Value from Outer query to generate a result.

COMMON Table Expressions (CTE)

```
with  s1  as ( _  ,  ,  )
      S2  as (         )

Select _ _ _ _
_ _ _ ,
inner join S1
ON _ _ _
Full join S2
ON _ _ _ ;
```

Window function

RANK() OVER

OVER(PARTITION BY)

ROWS BETWEEN 1 and 5

PRECEDING

FOLLOWING

UNBOUNDED PRECEDING

UNBOUNDED FOLLOWING

CURRENT ROW