

### CREATE

```
CREATE TABLE City (
    Name varchar (10)
    PRIMARY KEY,
    Population int,
    Province_Name
    varchar (10),
    CONSTRAINT FOREIGN KEY
    Province_Name
    References Provin -
    ce( Name)
);
```

### INSERT

```
INSERT INTO City (Name,
    Population, Province_Name)
    Values ("Bur naby", 250 -
    000 ,"BC ");
```

### DELETE

```
DELETE FROM City
    WHERE Name = " Bur nab y";
```

### IN

```
SELECT * FROM share
    WHERE shrcode IN
    ('FC', 'AR ', 'SLG');
```

Find shrcode that are FC, AR, SLG

### LIKE

```
Find students with surname
begins with 'Mc'
SELECT * FROM student WHERE
    surname LIKE "Mc%";
Find students with surname end
with 'ith'
SELECT * FROM student WHERE
    surname LIKE "%ith";
Find students with surname
having 'ack' inside
SELECT * FROM student WHERE
    surname LIKE "%ack%";
```

### A1: REGEXP

```
SELECT COUNT(*)
    FROM film
    WHERE specia l_f eatures
    REGEXP 'Deleted
    Scenes |Behind the Scenes'
    AND length >
    (SELECT AVG(le ngth)
    FROM film
    WHERE specia l_f -
    eatures REGEXP 'Comme nta -
    ries');
```

Get the count of movies that have "deleted scenes" or "behind the scenes" and are longer than average running length of all movies "commentaries"

### GROUP BY, HAVING, ORDER BY

```
SELECT province, city,
    SUM(revenue) AS TotalSales
    FROM sales
    WHERE countr y=' Canada'
    GROUP BY province, city
    HAVING TotalSales > 100000
    ORDER BY province, TotalS ales;
```

Think about the order of processing the data, filtering row by row (WHERE), then aggregating (GROUP BY), then filtering (HAVING), then sorting (ORDER BY)

### A1: AVG

```
SELECT AVG(rental_rate)
    FROM film
    WHERE rental _rate >
    (SELECT AVG(re nta l_rate)
    FROM film);
```

Get the average of rental rate of movies whose rental rates are higher than average

### A1: LENGTH

```
SELECT COUNT(*)
    FROM film
    WHERE rating = 'NC-17'
    AND LENGTH (ti tle )-L ENG TH( -
    REP LAC E(t itle, ' ', '')) >=2;
```

Get the count of NC-17 rating movies that have at least 3 words in their title

### A1: LENGTH

```
SELECT *
    FROM film
    WHERE LENGTH (title) >
    (SELECT AVG(LE NGT H(f irs -
    t_name) + LENGTH (la st_ nam -
    e)+1)
    FROM actor);
```

Get the list of films whose titles are longer than the average length of names of all actors in the actor table

### Data Types

CHAR(x)	VARCHAR(x)	INTEGER	DEC(x,y)
Exact length	Maximum length	or INT	x: total number of digits; y: number of decimal places
Alphan- umeric	Alphanumeric	Numeri	Num

