**Queries**

**# Exception**

* select 289+189;

**#Short pick:**

* multiple words alias: use ‘ ’

**1. give unique result(anti of repeat): DISTINCT**

-SELECT DISTINCT author\_lname FROM books;(2 books by same author won’t count

-SELECT DISTINCT CONCAT(author\_fname,' ', author\_lname) FROM books;

-SELECT DISTINCT author\_fname, author\_lname FROM books;(make effect on both column together;means making full name distinct)

**2. ascending descending : ORDER BY(make ascending or descending order)**

- SELECT author\_lname FROM books ORDER BY author\_lname;( default ascending

OR

SELECT author\_lname FROM books ORDER BY author\_lname ASC;

- SELECT author\_lname FROM books ORDER BY author\_lname DESC;(in descending order

- SELECT author\_lname ,released\_year FROM books ORDER BY released\_year;

- SELECT title, author\_fname, author\_lname

FROM books ORDER BY 2;( order by author\_fname; 2nd select column

-SELECT title, author\_fname, author\_lname

FROM books ORDER BY 3;( order by author\_lname

-SELECT title, author\_fname, author\_lname

FROM books ORDER BY 1;( order by title

-SELECT title, author\_fname, author\_lname

FROM books ORDER BY 1 DESC;

-SELECT author\_fname, author\_lname FROM books

ORDER BY author\_lname, author\_fname;( at first does order by author\_lname then make further order by author\_fname

**3.make a limitation on select : LIMIT INT/ LIMIT INT,INT**

-SELECT title FROM books LIMIT 3;

-SELECT title, released\_year FROM books

ORDER BY released\_year DESC LIMIT 5;

OR

SELECT title, released\_year FROM books

ORDER BY released\_year DESC LIMIT 0,5;

-SELECT title, released\_year FROM books

ORDER BY released\_year DESC LIMIT 10,1;( gives the 11th number row !!!

-SELECT title FROM books LIMIT 5, 123219476457;( show 5 to end

OR

SELECT title FROM books LIMIT 5, 500;

**4.searching with a matching: LIKE ‘%xy%’(for all) \‘%XY’(END WITH XY) \’XY%’(START WITH XY),**

**case insensitive,**

**sign %% is called WILDCARDS**

**-**SELECT title, author\_fname FROM books WHERE author\_fname LIKE '%da%';

**-**SELECT title, author\_fname FROM books WHERE author\_fname LIKE 'da%';( author\_fname start with da)

**-**SELECT title, author\_fname FROM books WHERE author\_fname LIKE 'da%';( author\_fname end with da)

-SELECT title FROM books WHERE title LIKE 'the';(out:empty set; we have no book which name is the

-SELECT title FROM books WHERE title LIKE '%the';(book title that start with the………ex: THE namesake,THE Halloween etc)

-SELECT title FROM books WHERE title LIKE '%the%';(out : **THE** namesake,O**THE**TT,OBI THE RTC; anywhere in the title where there is a the

-SELECT title FROM books WHERE title LIKE '%\%%';(for a % in the title,use \% ex: ‘10% happier’)

-SELECT title FROM books WHERE title LIKE '%\\_%';(for a \_ in the title, use \\_ ex:

‘FAKE\_BOOK’)

--SELECT title FROM books WHERE title LIKE '%%';(will select all entry(here all title))

Or

-SELECT title FROM books WHERE title LIKE '%%%%%';

Or

-SELECT title FROM books WHERE title LIKE '%';

(Another WILDDCARDS : \_)

-SELECT stock\_quantity FROM books WHERE stock\_quantity LIKE '\_\_\_\_';(1 \_ means 1 character or digit,here 4\_ means 1220,1000,1213.not 10022 or 22,222

-SELECT title, stock\_quantity FROM books WHERE stock\_quantity LIKE '\_\_';(EX : 12,10,99,’CH’

- (235)234-0987 = '(\_\_\_)\_\_\_-\_\_\_\_'

**AGGREGATE FUNCTION**

**5.COUNT(): COUNTING NUMBER OF ROWS EXIST IN A TABLE**

-SELECT COUNT(\*) FROM books; >COUNT HOW MANY ROWS(TOTAL) IN A TABLE

-SELECT COUNT(author\_fname) FROM books; > HOW MANY AUTHER\_FNAME WE HAVE

- SELECT COUNT( DISTINCT author\_fname) FROM books; > HOW MANY DISTINCT AUTHER\_FNAME WE HAVE(AS WE MIGHT HAVE MORE THAN 1 BOOKS FOR SAME WRITTER)

-SELECT COUNT(DISTINCT author\_lname) FROM books;

-SELECT COUNT(DISTINCT author\_lname, author\_fname) FROM books; >check if distinct by both column

- SELECT COUNT(\*) FROM books WHERE title LIKE '%the%';

**6.GROUP BY**

**Group by summarizes or aggregates data into single rows on those data which are possible to present in a single row.**

**Make grouping**

**-** SELECT released\_year, COUNT (\*) FROM books GROUP BY released\_year;

> each year how many books released

- SELECT author\_fname, author\_lname, COUNT (\*) FROM books GROUP BY author\_lname, author\_fname;

> how many authors are there and their names

**-** SELECT CONCAT ('In ', released\_year, ' ', COUNT(\*), ' book(s) released') FROM books GROUP BY released\_year;

>ex: IN 2013 3 book(s) released

- SELECT CONCAT ('In ', released\_year, ' ', COUNT(\*), ' book(s) released') as 'Release Info. 'FROM books GROUP BY released\_year;

* **Using multiple colum to group:**
* SELECT author\_fname, author\_lname, COUNT(\*)

FROM books

GROUP BY author\_lname, author\_fname;

* SELECT CONCAT(author\_fname, ' ', author\_lname) AS author, COUNT(\*)

FROM books

GROUP BY author; //author=the alias

* SELECT author\_lname, MIN(released\_year) FROM books GROUP BY author\_lname;
* SELECT author\_lname, MAX(released\_year), MIN(released\_year) FROM books GROUP BY author\_lname;
* SELECT

author\_lname,

COUNT(\*) as books\_written,

MAX(released\_year) AS latest\_release,

MIN(released\_year) AS earliest\_release,

MAX(pages) AS longest\_page\_count

FROM books GROUP BY author\_lname;

* SELECT

author\_lname,

author\_fname,

COUNT(\*) as books\_written,

MAX(released\_year) AS latest\_release,

MIN(released\_year) AS earliest\_release

FROM books GROUP BY author\_lname, author\_fname;

* **Sum**

***-****SELECT SUM(pages) FROM books;*

*-SELECT author\_lname, COUNT(\*), SUM(pages)*

*FROM books*

*GROUP BY author\_lname;*

* **Average**

-SELECT AVG(pages) FROM books;

-SELECT AVG(released\_year) FROM books;

-SELECT

released\_year,

AVG(stock\_quantity),

COUNT(\*) FROM books

GROUP BY released\_year;

**7. SUB QUERIES: “A subquery is a query that appears inside another query statement. Subqueries are also referred to as sub- SELECT s or nested SELECT s.”**

-SELECT title, pages FROM books

WHERE pages = (SELECT MAX(pages) FROM books);

-SELECT title, released\_year FROM books

WHERE released\_year = (SELECT MIN(released\_year) FROM books);