UNIT-2

Data Base Systems
SQL Relational Operations

Edited By: Er. Shilpi (AP, CSE)

# Logical Operators

- The Logical operators are those that are true or false. They return a true or false values to combine one or more true or false values.

The Logical operators are:

Operator	**Description**
AND	Logical AND compares between two Booleans as expression and returns true when both expressions are true...
OR	Logical OR compares between two Booleans as expression and returns true when one of the expression is true...
NOT	Not takes a single Boolean as an argument and changes its value from false to true or from true to false.

Special Operators: for Correlated Queries

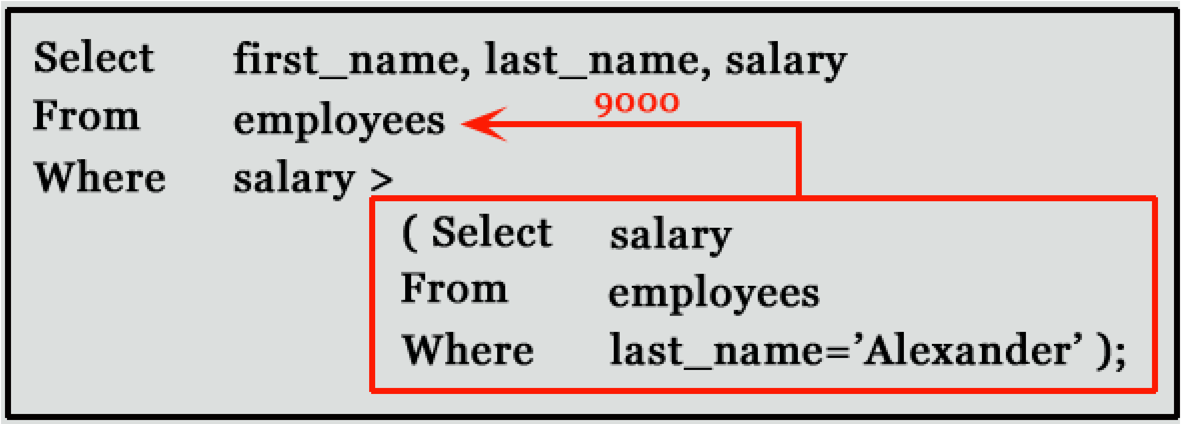
Operator	Description	Operates on
IN	The IN operator checks a value within a set of values separated by commas and retrieve the rows from the table which are matching....	Any set of values of the same datatype
BETWEEN	The SQL BETWEEN operator tests an expression against a range. The range consists of a beginning, followed by an AND keyword and an end expression....	Numeric, characters, or datetime values
ANY	ANY compares a value to each value in a list or results from a query and evaluates to true if the result of an inner query contains at least one row....	A value to a list or a single - columns set of values
ALL	ALL is used to select all records of a SELECT STATEMENT. It compares a value to every value in a list or results from a query. The ALL must be preceded by the comparison operators and evaluates to TRUE if the query returns no rows....	A value to a list or a single - columns set of values
SOME	SOME compare a value to each value in a list or results from a query and evaluate to true if the result of an inner query contains at least one row...	A value to a list or a single - columns set of values
EXISTS	The EXISTS checks the existence of a result of a subquery. The EXISTS subquery tests whether a subquery fetches at least one row. When no data is returned then this operator returns 'FALSE'...	Table

Correlated Queries or Subqueries:

- A subquery may occur in:
  - - A SELECT clause
  - - A FROM clause
  - - A WHERE clause
- In MySQL subquery can be nested inside a SELECT, INSERT, UPDATE, DELETE etc. statement or inside another subquery.
- A subquery is usually added within the WHERE Clause of another SQL SELECT statement.
- You can use the comparison operators, such as >, <, or =. The comparison operator can also be a multiple-row operator, such as IN, ANY, SOME, or ALL.
- A subquery can be treated as an inner query, which is a SQL query placed as a part of another query called as outer query.
- The inner query executes first before its parent query so that the results of the inner query can be passed to the outer query.

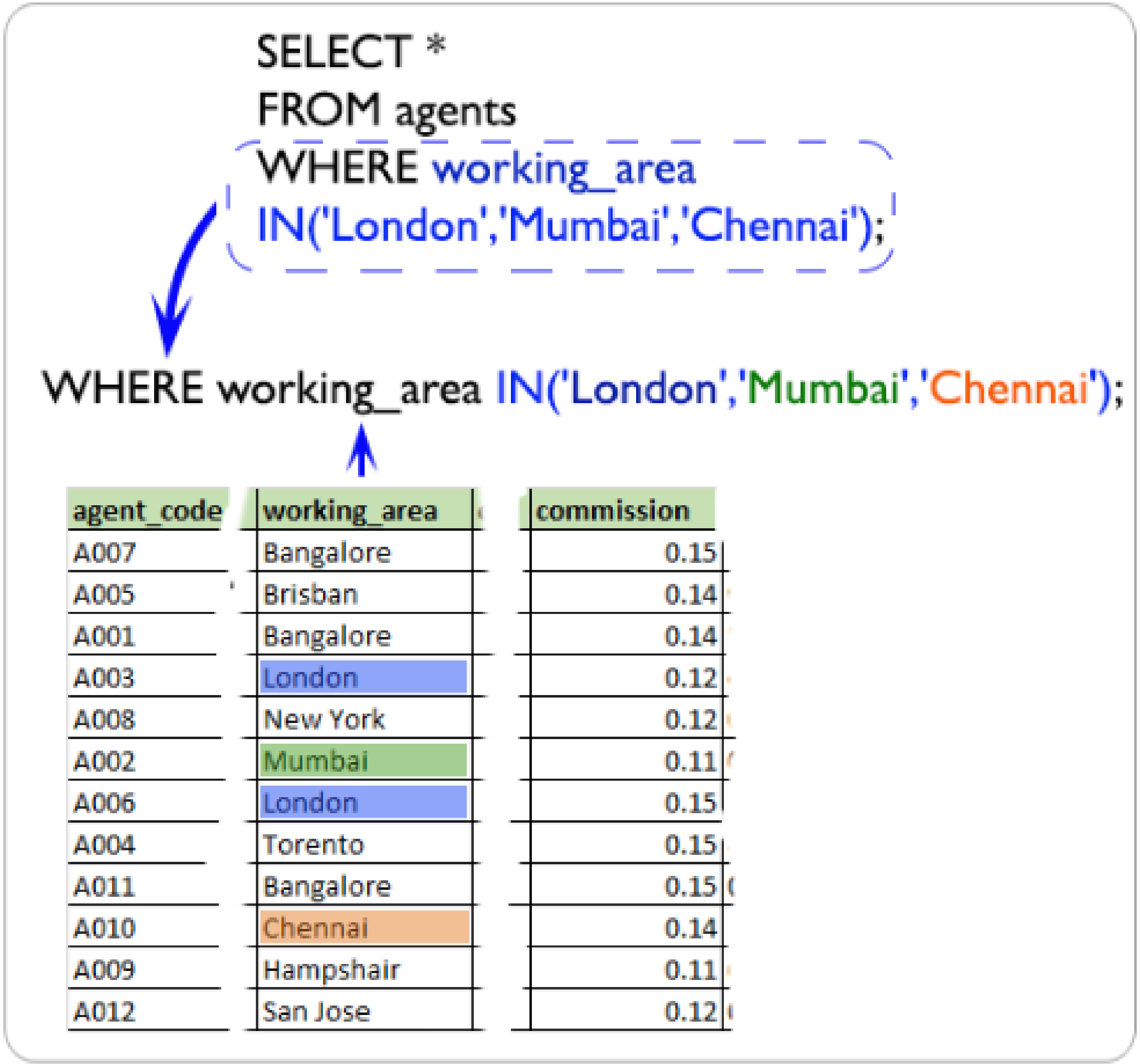
Correlated Queries or Subqueries:

- The inner query executes first before its parent query so that the results of the inner query can be passed for the every record of outer query.



```
Select      first_name, last_name, salary
From        employees
Where       salary >
                  ( Select     salary
                    From       employees
                    Where      last_name='Alexander' );
```

IN Operator



```
SELECT *
FROM agents
WHERE working_area
IN('London','Mumbai','Chennai');
```

```
WHERE working_area IN('London','Mumbai','Chennai');
```

agent_code	working_area	commission
A007	Bangalore	0.15
A005	Brisban	0.14
A001	Bangalore	0.14
A003	London	0.12
A008	New York	0.12
A002	Mumbai	0.11
A006	London	0.15
A004	Torento	0.15
A011	Bangalore	0.15
A010	Chennai	0.14
A009	Hampshair	0.11
A012	San Jose	0.12

IN Operator

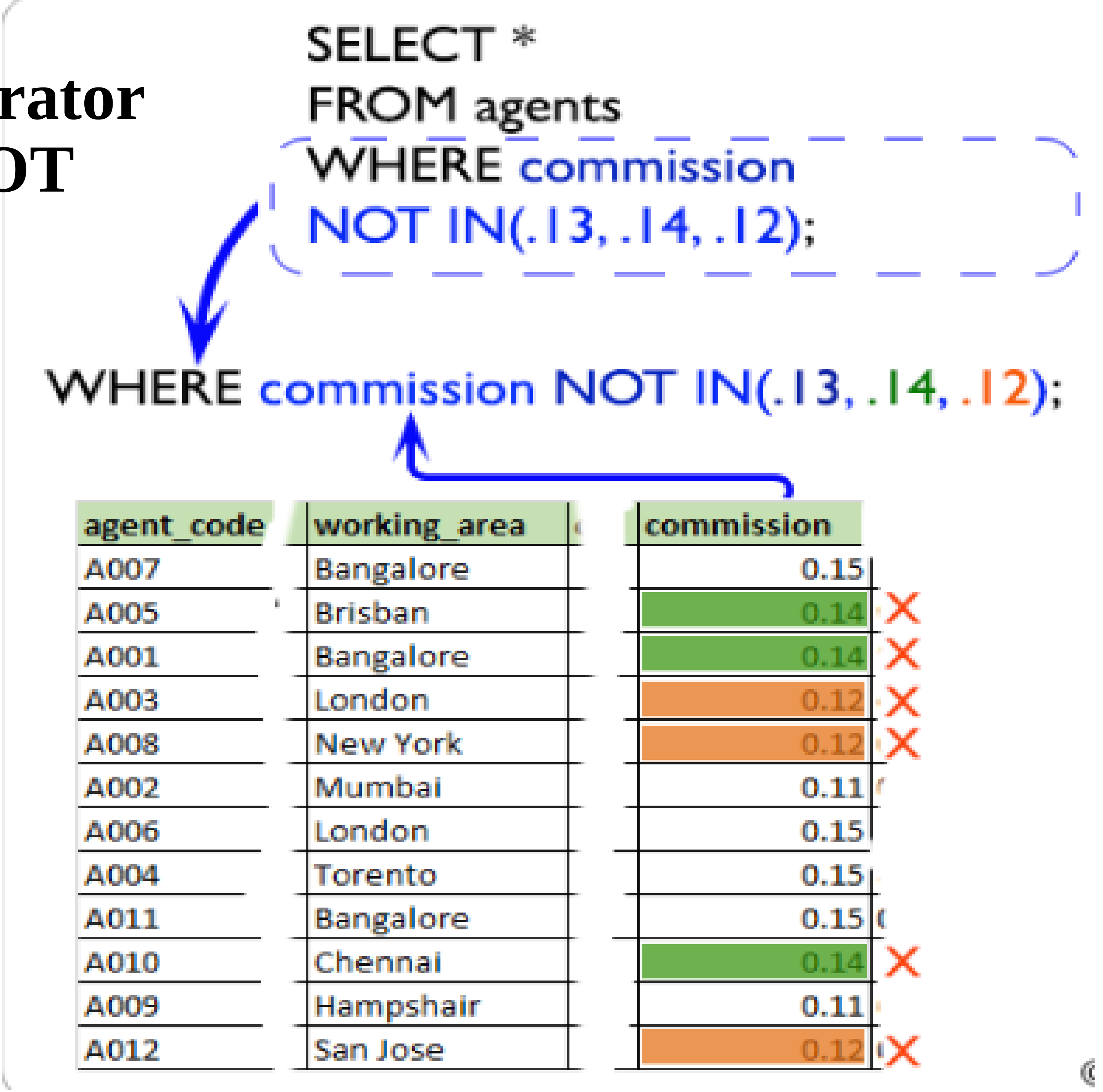
- The checking value of IN operator can be a string or word or sentence. These values can also be checked within a set of values separated by commas and retrieve the rows containing these values.

```
>>> SELECT * FROM agents WHERE working_area
      IN ( 'London‘ , 'Mumbai‘ , 'Chennai' );
```

This statement can also be used like bellow:

```
>>> SELECT * FROM agents WHERE
working_area='London' OR working_area='Mumbai'
OR working_area='Chennai';
```

IN Operator with NOT

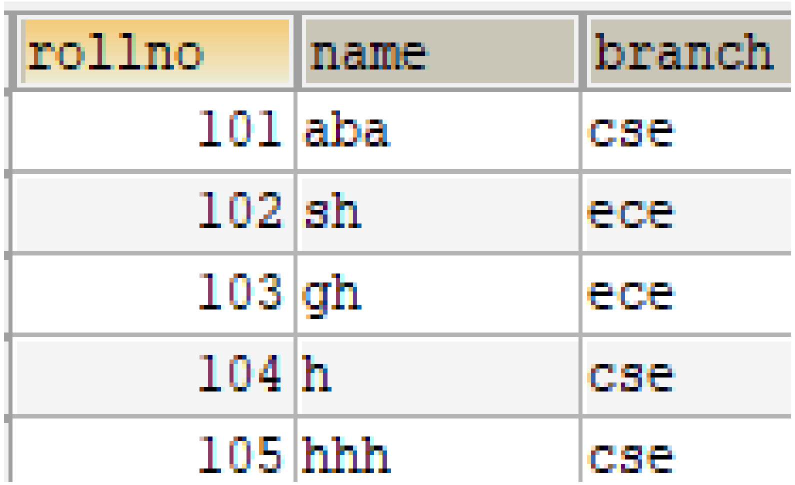


```
SELECT *
FROM agents
WHERE commission
NOT IN(.13, .14, .12);
```

```
WHERE commission NOT IN(.13, .14, .12);
```

agent_code	working_area	commission	
A007	Bangalore	0.15	
A005	Brisban	0.14	X
A001	Bangalore	0.14	X
A003	London	0.12	X
A008	New York	0.12	X
A002	Mumbai	0.11	
A006	London	0.15	
A004	Torento	0.15	
A011	Bangalore	0.15	
A010	Chennai	0.14	X
A009	Hampshair	0.11	
A012	San Jose	0.12	X

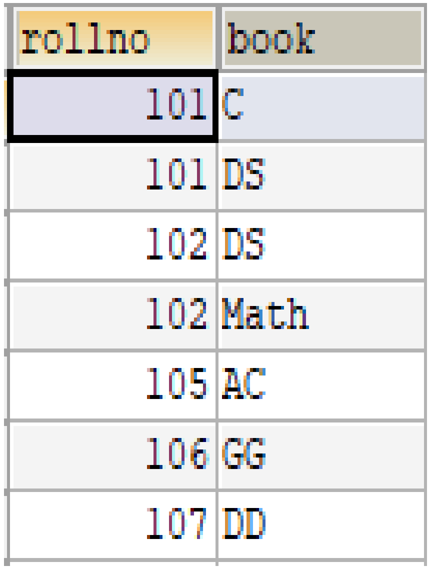
IN Operator



rollno	name	branch
101	aba	cse
102	sh	ece
103	gh	ece
104	h	cse
105	hhh	cse

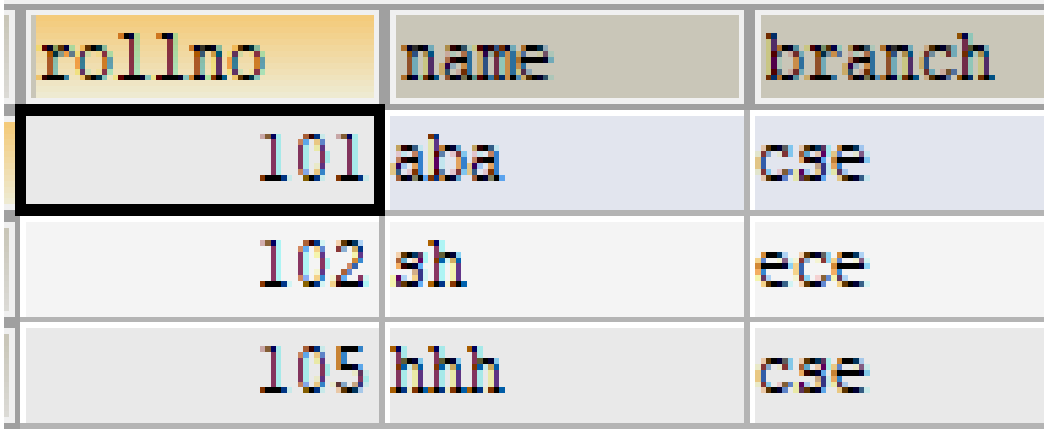
<- Stuinfo

Stulib->



rollno	book
101	C
101	DS
102	DS
102	Math
105	AC
106	GG
107	DD

>>>SELECT *FROM stuinfo WHERE rollno IN
(SELECT rollno FROM stulib);

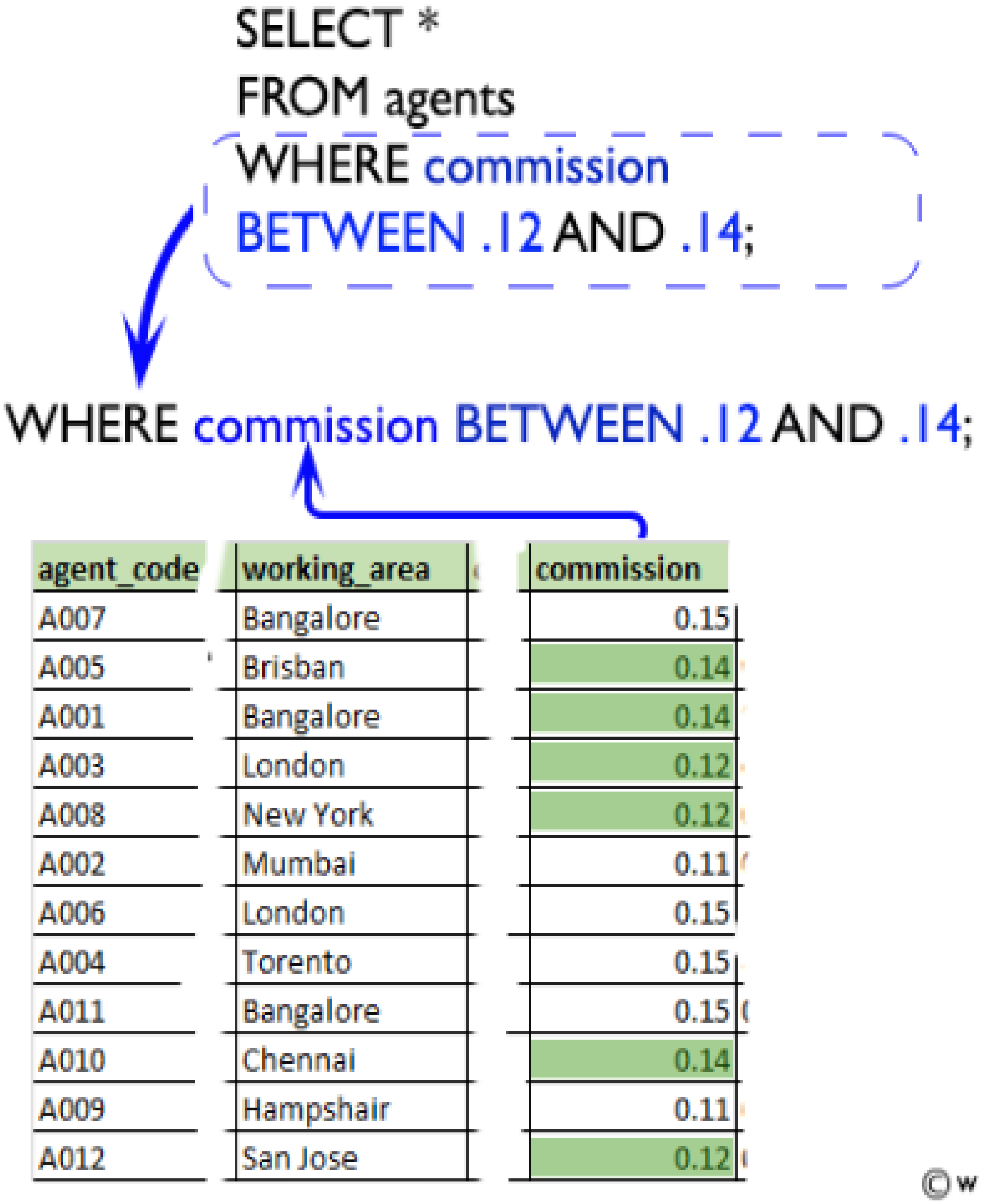


rollno	name	branch
101	aba	cse
102	sh	ece
105	hhh	cse

Between Operator

- The SQL BETWEEN operator tests an expression against a range. The range consists of a beginning, followed by an AND keyword and an end expression. The operator returns TRUE when the search value present within the range otherwise returns FALSE. The results are NULL if any of the range values are NULL.

>>>SELECT * FROM agents WHERE commission BETWEEN .12 AND .14;

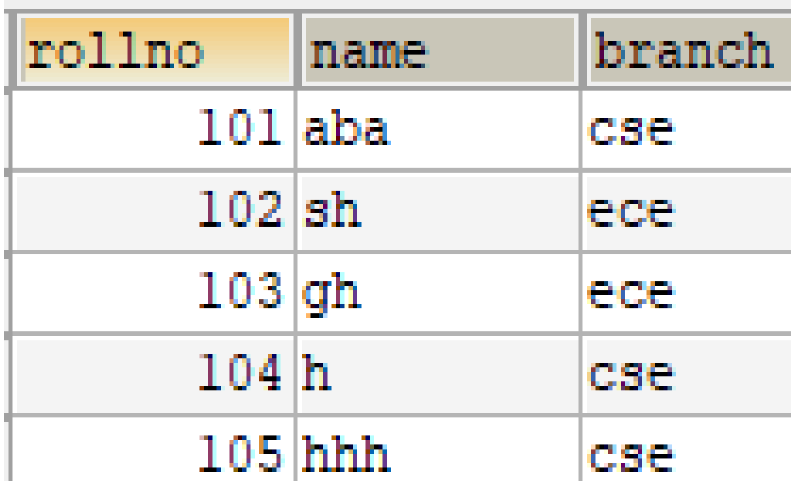


```
SELECT *
FROM agents
WHERE commission
BETWEEN .12 AND .14;
```

```
WHERE commission BETWEEN .12 AND .14;
```

agent_code	working_area	commission
A007	Bangalore	0.15
A005	Brisban	0.14
A001	Bangalore	0.14
A003	London	0.12
A008	New York	0.12
A002	Mumbai	0.11
A006	London	0.15
A004	Torento	0.15
A011	Bangalore	0.15
A010	Chennai	0.14
A009	Hampshair	0.11
A012	San Jose	0.12

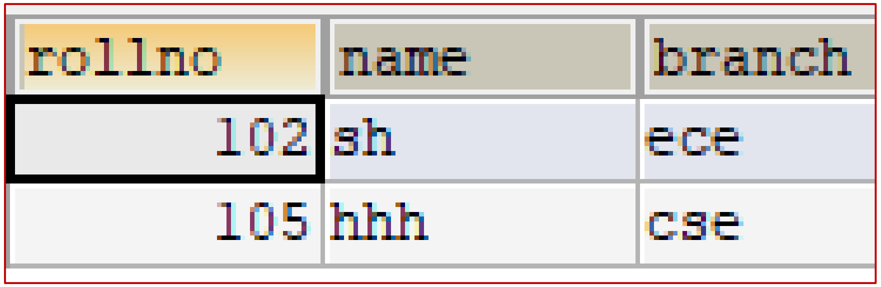
BETWEEN Operator



rollno	name	branch
101	aba	cse
102	sh	ece
103	gh	ece
104	h	cse
105	hhh	cse

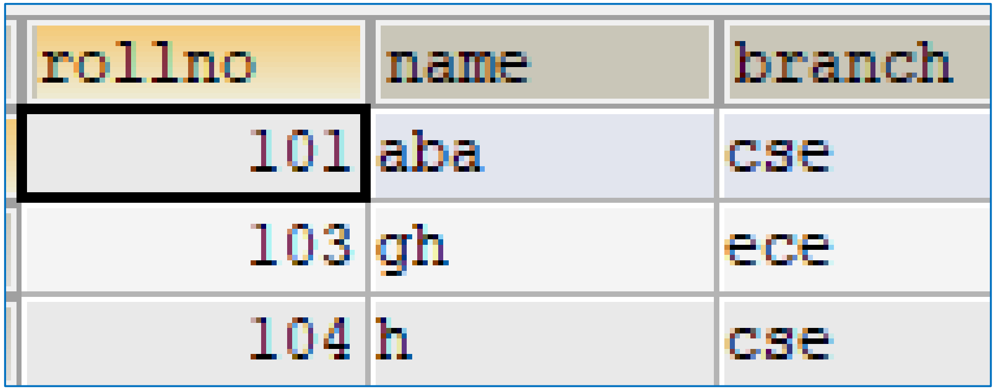
<- Stuinfo

>SELECT *FROM stuinfo WHERE NAME NOT BETWEEN 'a' AND 'h';



rollno	name	branch
102	sh	ece
105	hhh	cse

>SELECT *FROM stuinfo WHERE NAME BETWEEN 'a' AND 'h';



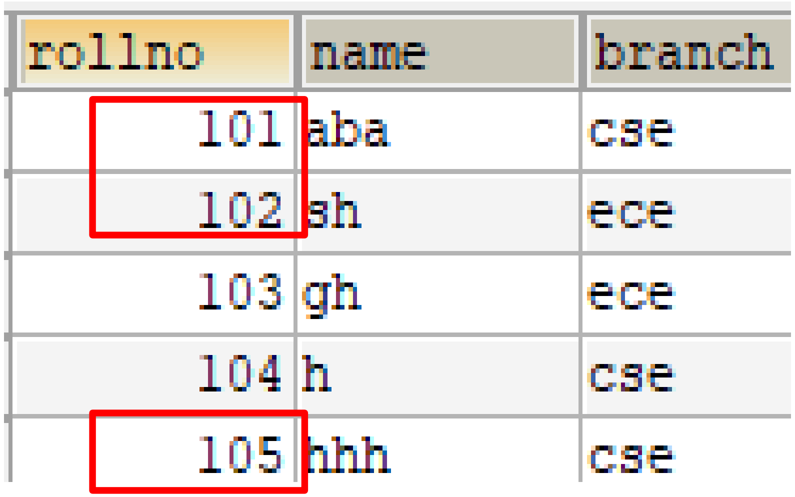
rollno	name	branch
101	aba	cse
103	gh	ece
104	h	cse

ANY Operator

- ANY compares a value to each value in a list or results from a query and evaluates to true if the result of an inner query contains at least one row. ANY must be preceded by comparison operators. Suppose using greater than (>) with ANY means greater than at least one value OR using less than (<) with ANY value etc.

```
>>>SELECT agent_code, agent_name, working_area,
commission FROM  agents WHERE
agent_code = ANY (SELECT agent_code FROM
customer WHERE cust_country='UK');
```

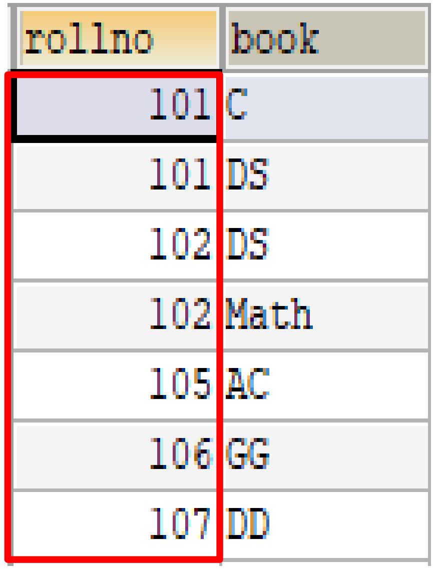
ANY Operator



rollno	name	branch
101	aba	cse
102	sh	ece
103	gh	ece
104	h	cse
105	hhh	cse

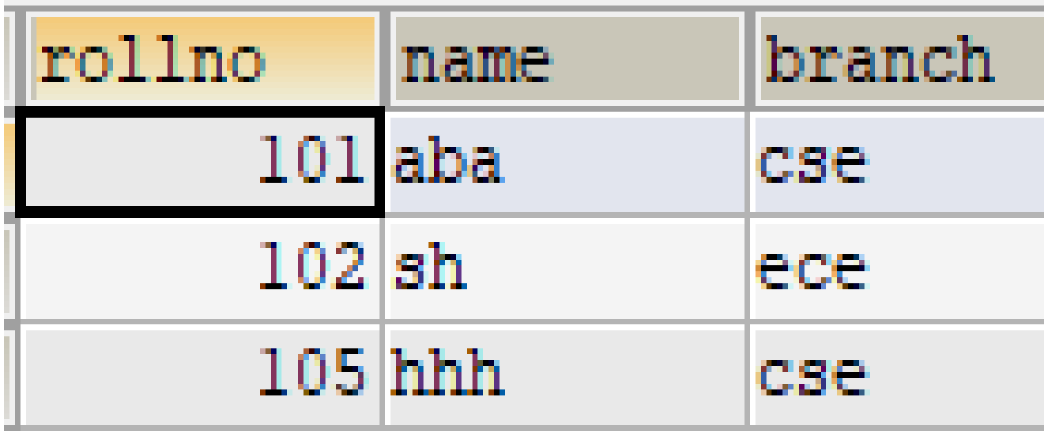
<- Stuinfo

Stulib->



rollno	book
101	C
101	DS
102	DS
102	Math
105	AC
106	GG
107	DD

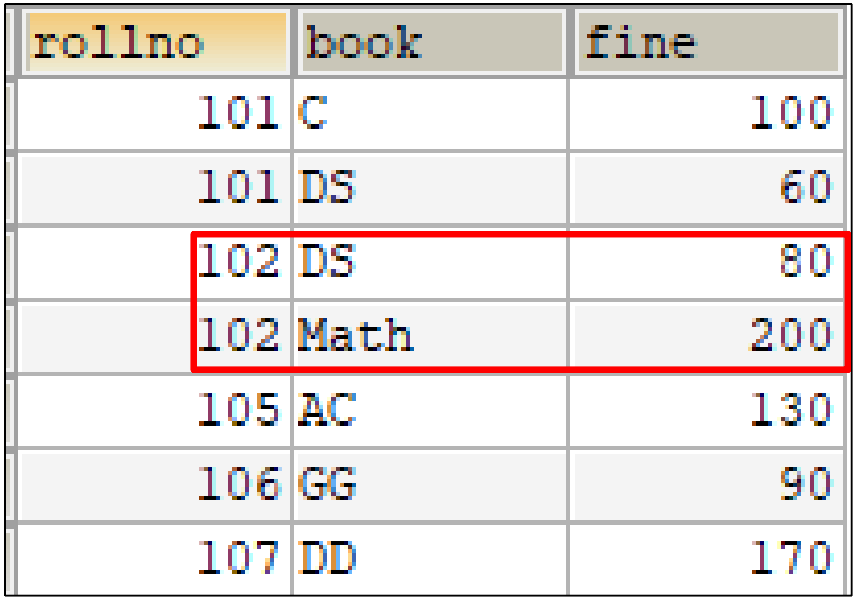
>>>SELECT *FROM stuinfo WHERE rollno = ANY
(SELECT rollno FROM stulib);



rollno	name	branch
101	aba	cse
102	sh	ece
105	hhh	cse

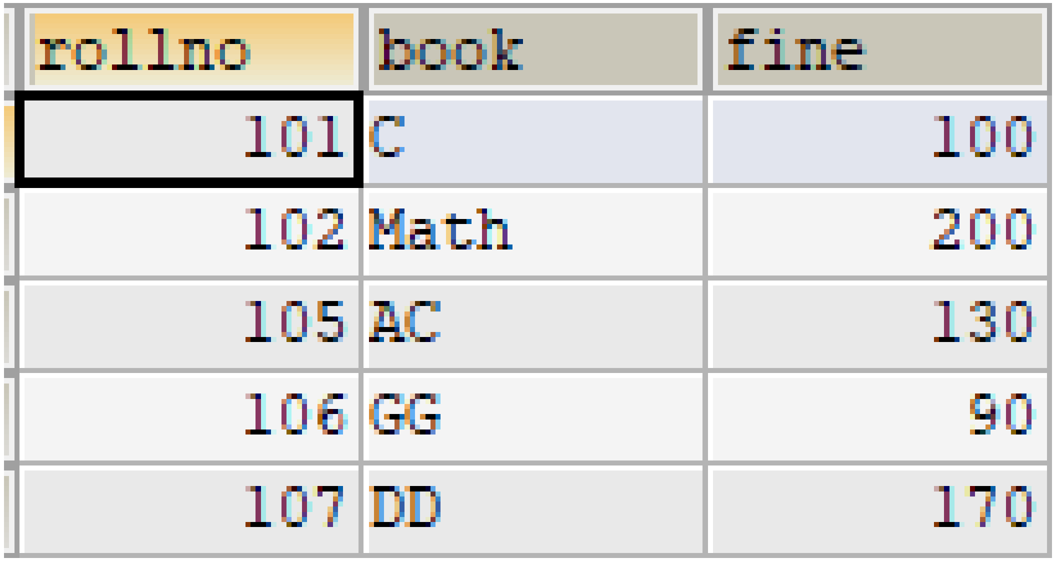
ANY Operator

Stulib->



rollno	book	fine
101	C	100
101	DS	60
102	DS	80
102	Math	200
105	AC	130
106	GG	90
107	DD	170

>>> SELECT * FROM stulib WHERE fine > ANY (SELECT fine FROM stulib WHERE rollno=102);



rollno	book	fine
101	C	100
102	Math	200
105	AC	130
106	GG	90
107	DD	170

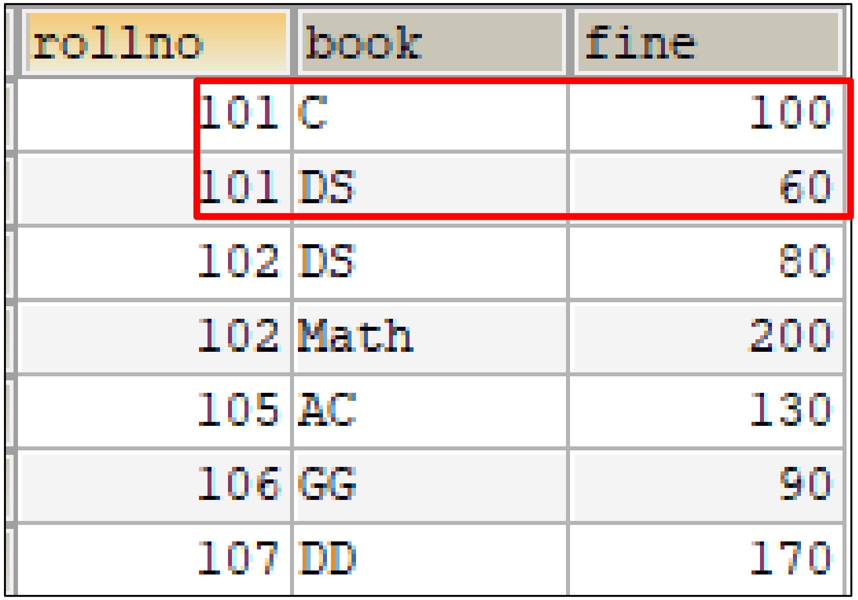
ALL Operator

- ALL is used to select all records of a SELECT STATEMENT. It compares a value to every value in a list or results from a query. The ALL must be preceded by the comparison operators and evaluates to FALSE if the query returns no rows. For example, ALL means greater than every value, means greater than the maximum value. Suppose ALL (1, 2, 3) means greater than 3.

```
>SELECT dis_date, dis_amount, ord_amount
    FROM dispatch WHERE
    dis_amount > ALL (SELECT ord_amount FROM
    orders WHERE ord_amount >= 2000);
```

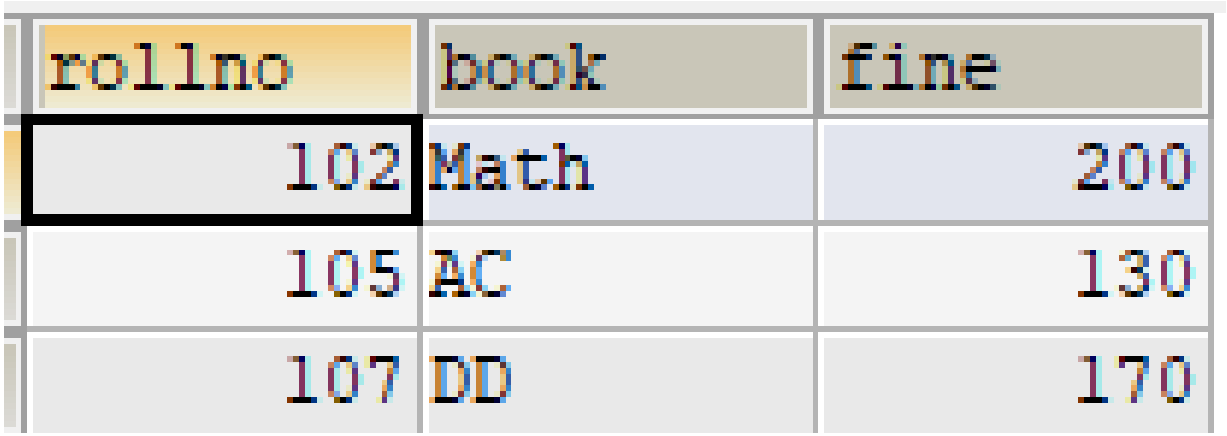
ALL Operator

Stulib->



rollno	book	fine
101	C	100
101	DS	60
102	DS	80
102	Math	200
105	AC	130
106	GG	90
107	DD	170

>>> SELECT * FROM stulib WHERE fine > ALL (SELECT fine FROM stulib WHERE rollno=101);



rollno	book	fine
102	Math	200
105	AC	130
107	DD	170

SOME Operator

- SOME compare a value to each value in a list or results from a query and evaluate to true if the result of an inner query contains at least one row. SOME must match at least one row in the subquery and must be preceded by comparison operators. Suppose using greater than (>) with SOME means greater than at least one value.
- It’s output is somehow similar to ANY operator.

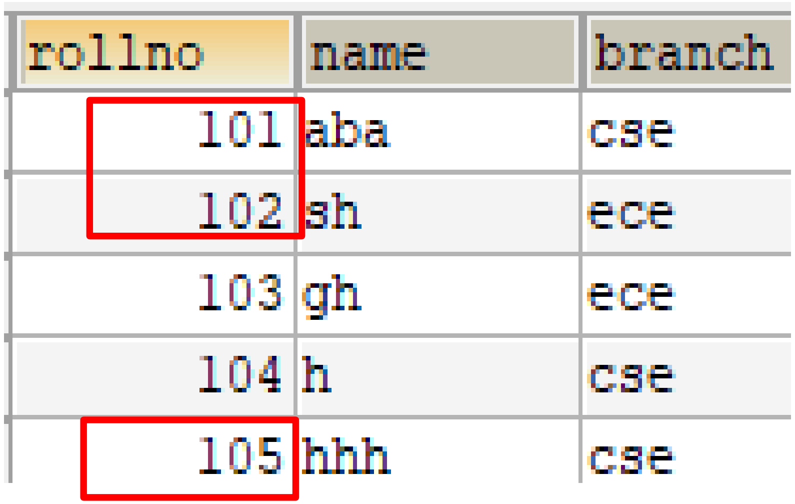
```
>>>SELECT agent_code, agent_name, working_area,
commission FROM  agents WHERE
agent_code = SOME (SELECT agent_code FROM
customer WHERE cust_country='UK');
```

EXISTS Operator

- The EXISTS checks the existence of a result of a Subquery. The EXISTS subquery tests whether a subquery fetches at least one row. When no data is returned then this operator returns 'FALSE'. A valid EXISTS subquery must contain an outer reference and it must be a correlated Subquery. The select list in the EXISTS subquery is not actually used in evaluating the EXISTS so it can contain any valid select list.

```
>SELECT agent_code, agent_name, working_area, commission FROM agents WHERE exists (SELECT * FROM customer WHERE grade=3 AND agents.agent_code = customer.agent_code);
```

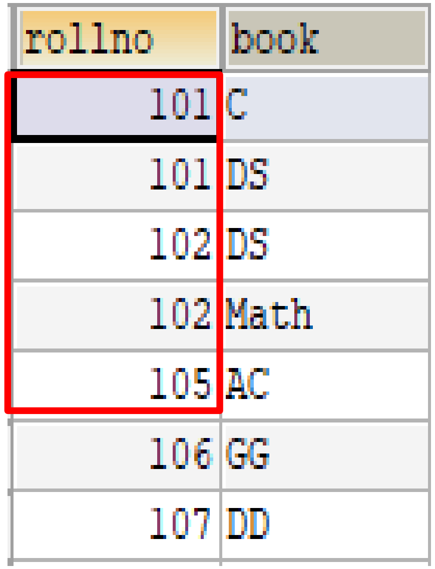
EXISTS Operator



rollno	name	branch
101	aba	cse
102	sh	ece
103	gh	ece
104	h	cse
105	hhh	cse

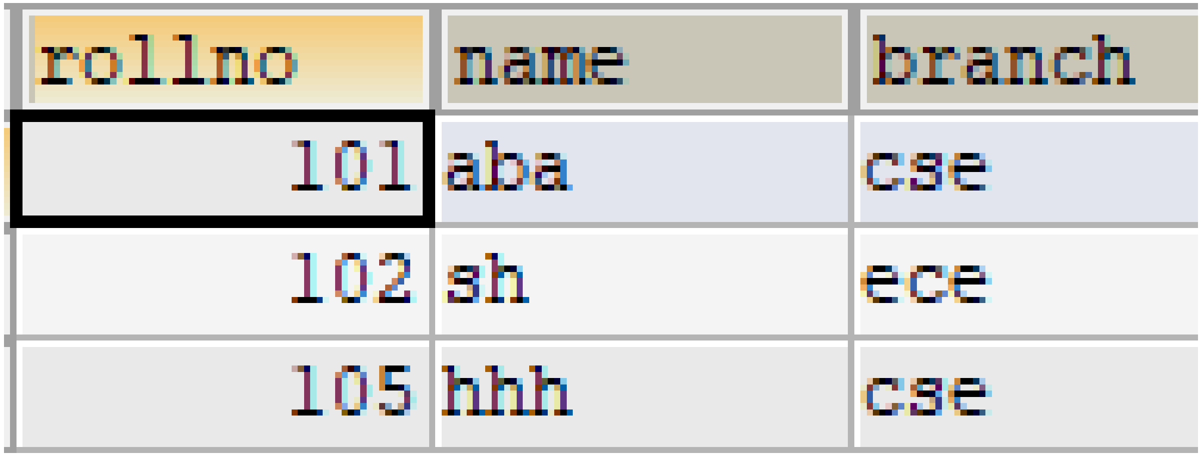
<- Stuinfo

Stulib->



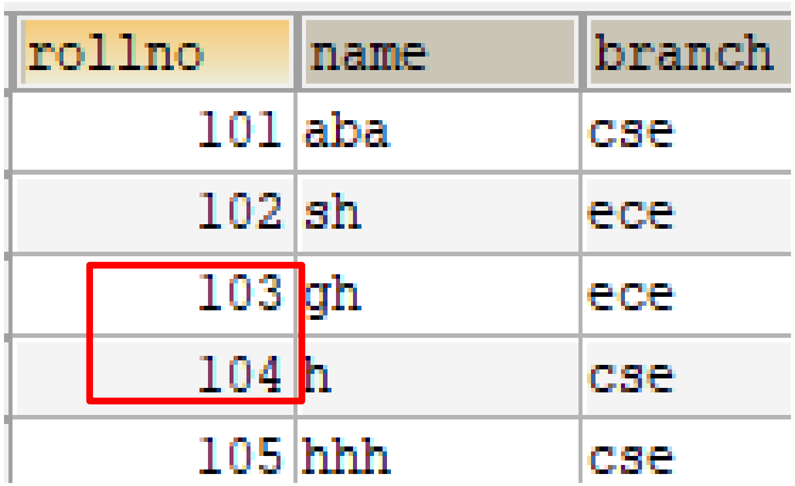
rollno	book
101	C
101	DS
102	DS
102	Math
105	AC
106	GG
107	DD

```
>>>SELECT * FROM stuinfo WHERE EXISTS (SELECT
rollno FROM stulib WHERE stuinfo.rollno=stulib.rollno);
```



rollno	name	branch
101	aba	cse
102	sh	ece
105	hhh	cse

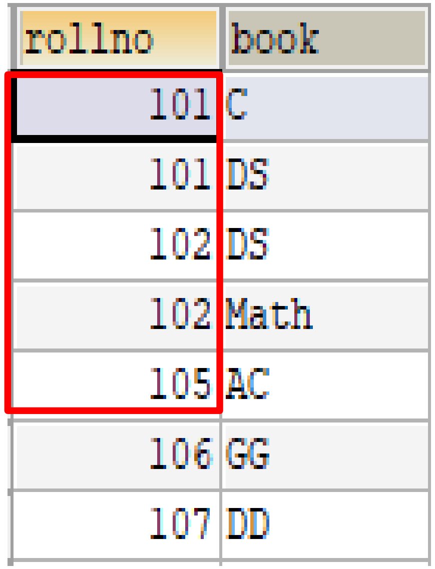
EXISTS Operator



rollno	name	branch
101	aba	cse
102	sh	ece
103	gh	ece
104	h	cse
105	hhh	cse

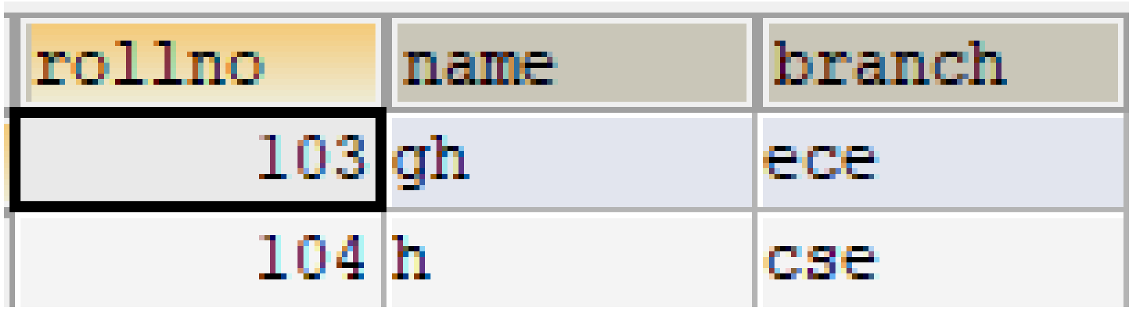
<- Stuinfo

Stulib->



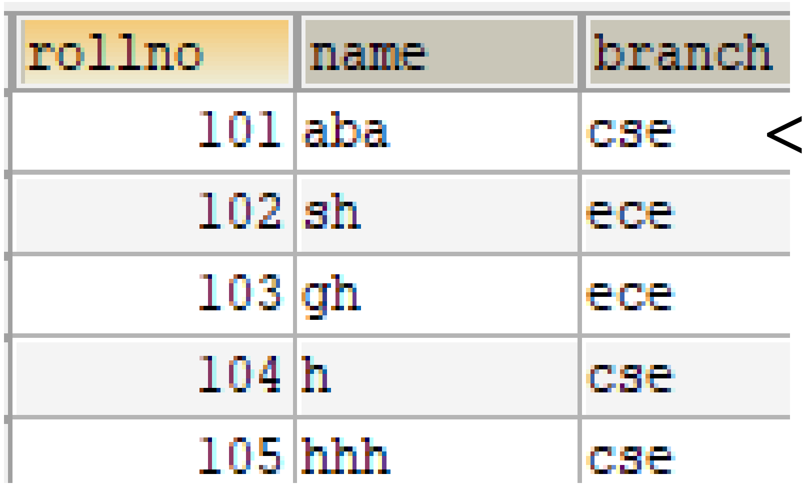
rollno	book
101	C
101	DS
102	DS
102	Math
105	AC
106	GG
107	DD

```
>>>SELECT * FROM stuinfo WHERE NOT EXISTS
(SELECT rollno FROM stulib WHERE
stuinfo.rollno=stulib.rollno);
```



rollno	name	branch
103	gh	ece
104	h	cse

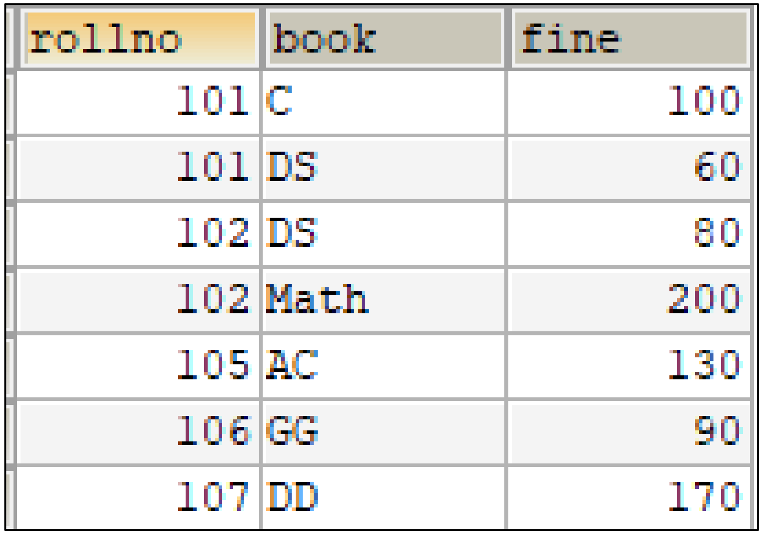
EXISTS Operator



rollno	name	branch
101	aba	cse
102	sh	ece
103	gh	ece
104	h	cse
105	hhh	cse

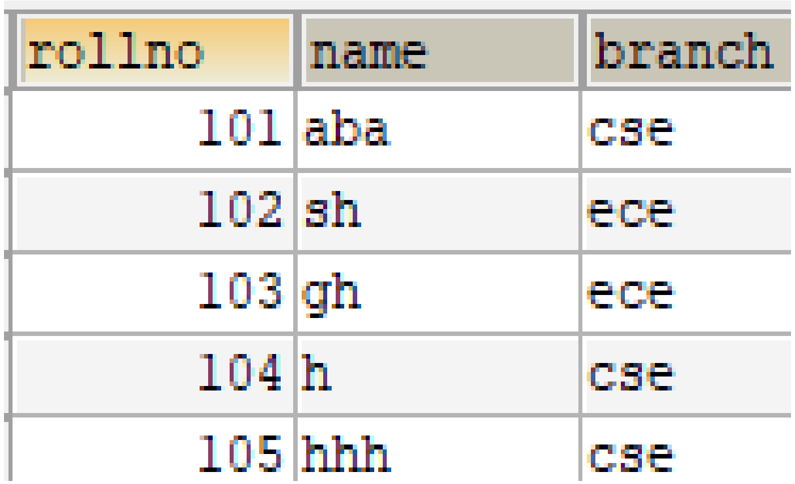
<-Stuinfo

Stulib->



rollno	book	fine
101	C	100
101	DS	60
102	DS	80
102	Math	200
105	AC	130
106	GG	90
107	DD	170

>>>SELECT * FROM stuinfo WHERE EXISTS (SELECT rollno FROM stulib WHERE stuinfo.rollno != stulib.rollno);



rollno	name	branch
101	aba	cse
102	sh	ece
103	gh	ece
104	h	cse
105	hhh	cse

EXISTS Operator

```
>SELECT * FROM stuinfo WHERE NOT EXISTS(SELECT rollno FROM stulib);
```

//No resulting rows

```
>SELECT * FROM stulib WHERE EXISTS(SELECT fine FROM stulib WHERE rollno=101);
```

// complete stulib table