

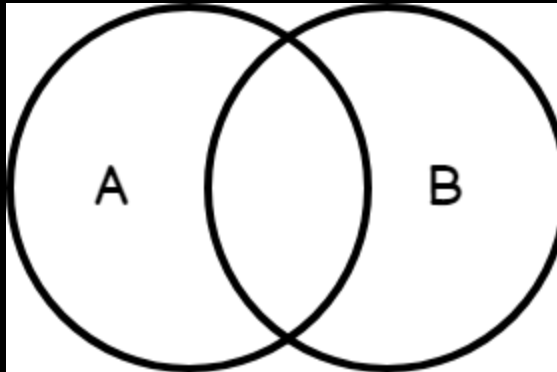


CHAPTER 9

ADVANCE DATA MANIPULATION LANGUAGE

9.1 UNION

ពាក្យបញ្ជា UNION រួមបញ្ចូលគ្នានូវសំណុំលទ្ធផលនៃ SELECT Statement ពីរប្រើន (មានតែតម្លៃខុសគ្នាប៉ុណ្ណោះ)។



A	B		A UNION B
1	3	→	1
2	4		2
3	5		3
			4
			5

EXAMPLE:

Table: Art_Student

Student_Number	Student_Name	Student_Marks
1	John	95
2	Mary	80
3	Damon	57

Table: Dance_Student

Student_Number	Student_Name	Student_Marks
2	Mary	50
3	Damon	98
6	Matt	45

CONT'D

Select Student_Name from Art_Student

UNION

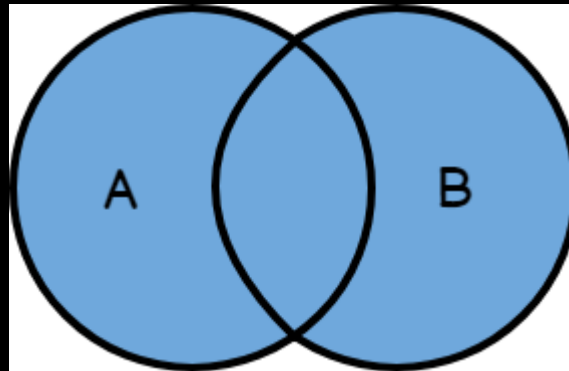
Select Student_Name from Dance_Student

Result:

Student_Name
John
Mary
Damon
Matt

9.2 UNION ALL

បើត្រូវឡើងមើល Union vs Union All អ្វីដែលយើងរកឃើញថាពួកគេមានលក្ខណៈ
ប្រហាក់ប្រហែលគ្នាប៉ុន្តែពួកគេមានភាពខុសគ្នាសំខាន់ៗ។



A	B		A UNION ALL B
1	3	→	1
2	4		2
3	5		3
			3
			4
			5

EXAMPLE:

Select Student_Name from Art_Student

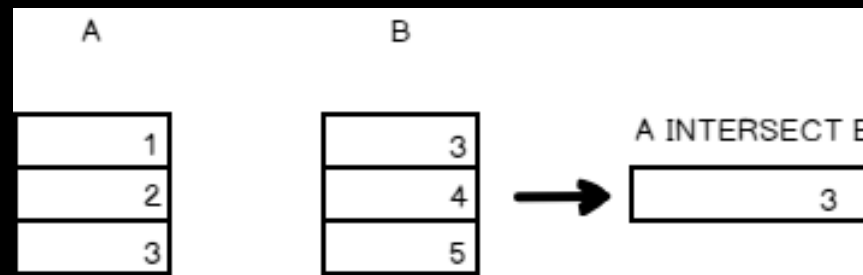
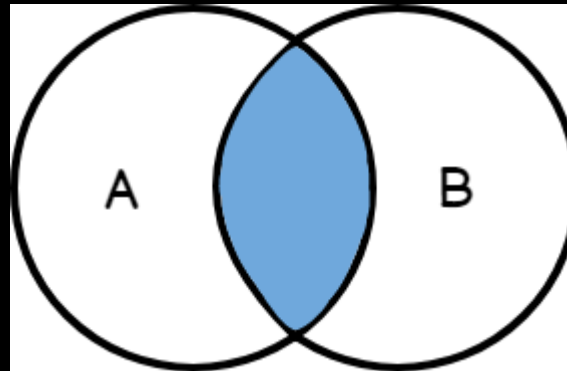
UNION ALL

Select Student_Name from Dance_Student

Student_Name
John
Mary
Damon
Mary
Damon
Matt

9.3 INTERSECT

INTERSECT ត្រូវបានប្រើដើម្បីត្រឡប់លទ្ធផលនៃ SELECT Statement ចំនួន ២ រឺច្រើន។ ទោះយ៉ាងណាក៏ដោយវាត្រឡប់មកវិញតែជួរដេកទាំងឡាយណាដែល Table ទាំងមានដូចគ្នា។



EXAMPLE:

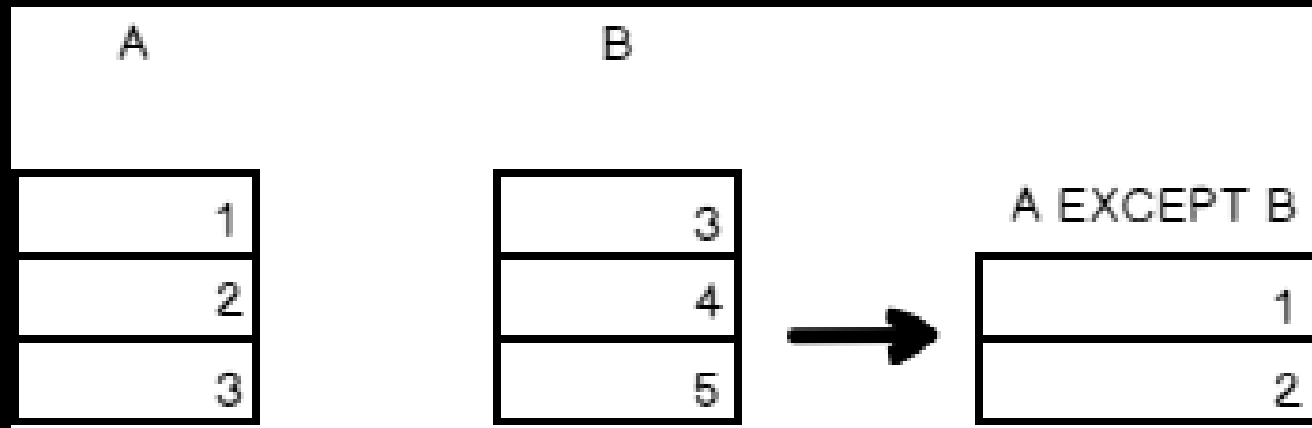
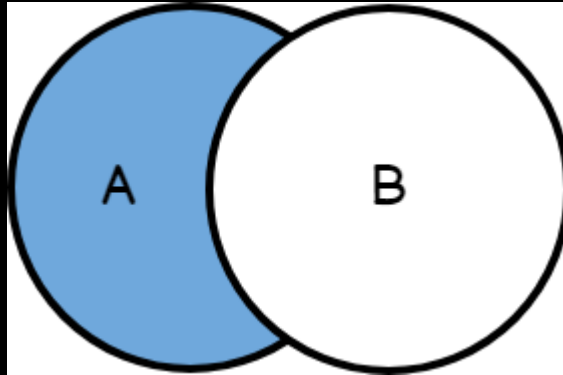
Select Student_Name from Art_Student

INTERSECT

Select Student_Name from Dance_Student

Student_Name
Mary
Damon

9.4 EXCEPT



EXAMPLE:

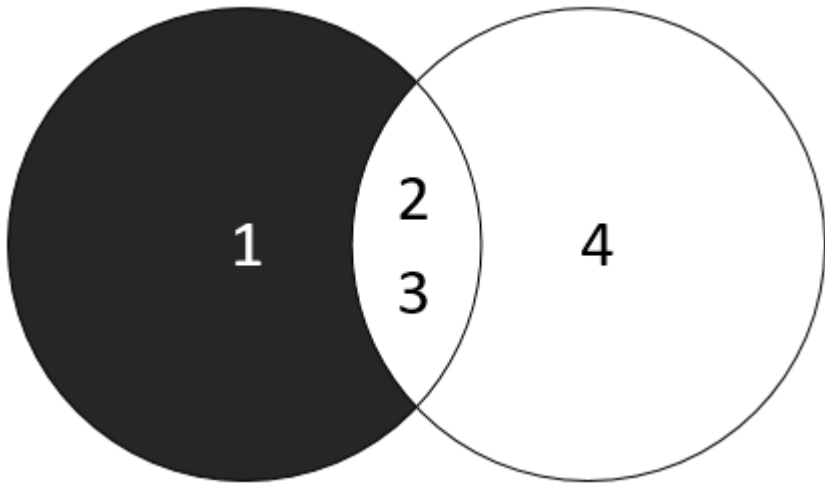
Select Student_Name from Art_Student

EXCEPT

Select Student_Name from Dance_Student

Student_Name
John

9.5 MINUS



```
SELECT id FROM t1  
MINUS  
SELECT id FROM t2;
```

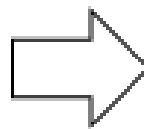
id
1
2
3

t1 table

MINUS

id
2
3
4

t2 table

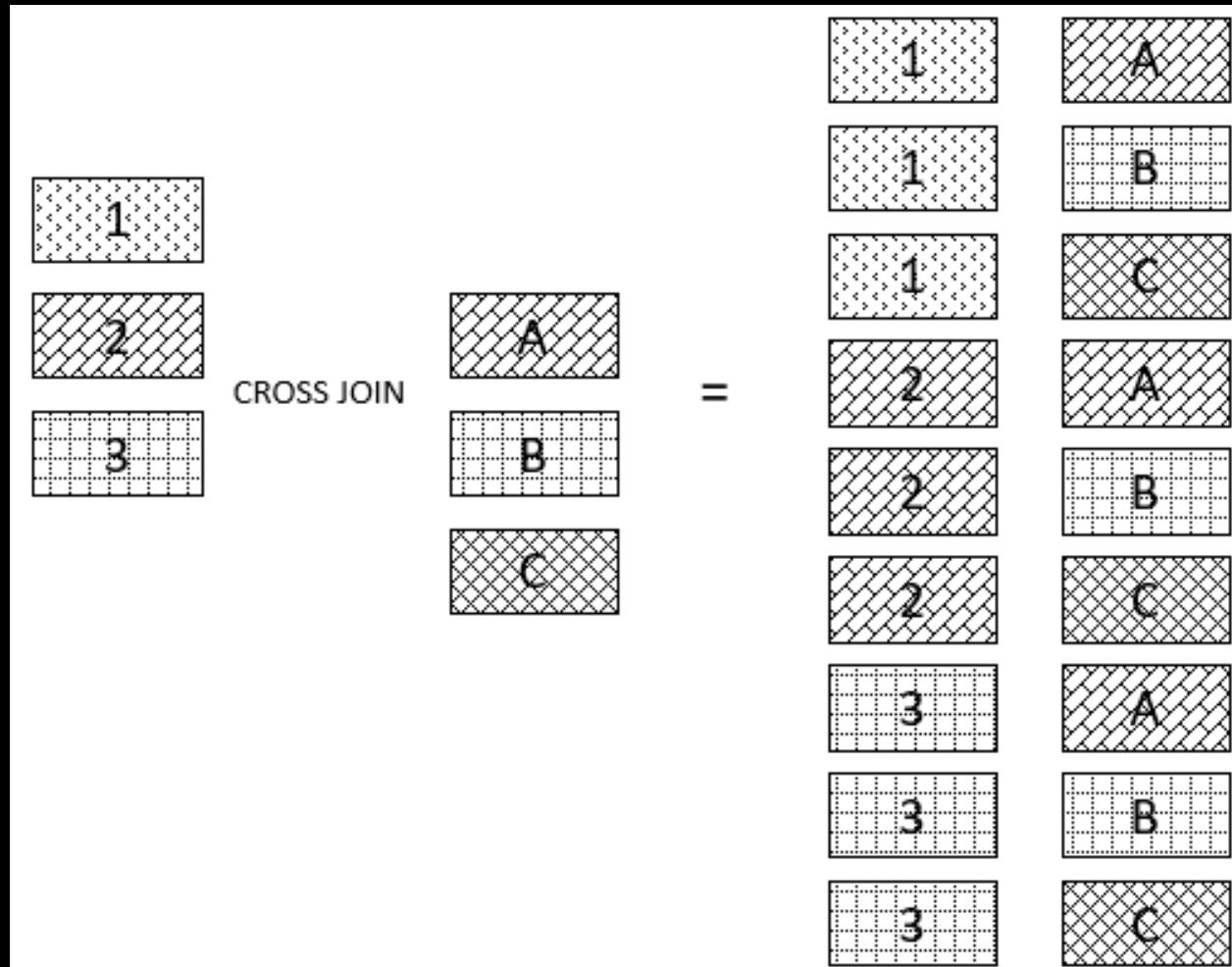


id
1

```
SELECT id FROM t1  
LEFT JOIN t2 USING (id)  
WHERE t2.id IS NULL;
```

9.4 CROSS JOIN

CROSS JOIN បានចូលរួមគ្រប់ជួរពីតារាងទីមួយ (T1) ជាមួយជួរទាំងអស់ពីតារាងទីពីរ (T2) ។



EXAMPLE:

Table: foods

Item_ID	Item_Name	Item_Unit	Company_ID
1	Chex Mix	Pcs	16
6	Cheez-it	Pcs	15
2	BN Biscuit	Pcs	15
3	Mighty Munch	Pcs	17
4	Pot Rice	Pcs	15
5	Jaffa Cakes	Pcs	18
7	Salt n Shake	Pcs	NULL

Table: companies

Company_ID	Company_Na	Company_City
18	Order All	Boston
15	Jack Hill Ltd	London
16	Akas Foods	Delhi
17	Foodies	London
19	sip-n-Bite.	New York

```
SELECT foods.item_name,foods.item_unit,  
companies.company_name,companies.company_city  
FROM foods  
CROSS JOIN companies;
```

9.5 NATURAL JOIN

ធ្វើការតែនៅពេល Table 2 មានជួរឈរមានឈ្មោះនិងប្រភេទដូចគ្នាតែមួយដែលជាកូនសោចូលរួម។ Natural Join គឺជិតដូចគ្នានឹង INNER JOIN លើកលែងតែជួរឈរដដែលៗត្រូវបានជៀសវាង។ នៅពេលប្រើពាក្យ NATURAL JOIN ពាក្យគន្លឹះ ON និង USING មិនគួរត្រូវបានអនុញ្ញាត។ មិនមែនគ្រប់ RDBMS បានអនុវត្ត NATURAL JOIN ទេ។

EXAMPLE:

Table: Foods

Item_ID	Item_Name	Item_Unit	Company_ID
1	Chex Mix	Pcs	16
6	Cheez-it	Pcs	15
2	BN Biscuit	Pcs	15
3	Mighty Munch	Pcs	17
4	Pot Rice	Pcs	15
5	Jaffa Cakes	Pcs	18
7	Salt n Shake	Pcs	NULL

Table: Companies

Company_ID	Company_Na	Company_City
18	Order All	Boston
15	Jack Hill Ltd	London
16	Akas Foods	Delhi
17	Foodies	London
19	sip-n-Bite.	New York

```
SELECT *  
FROM foods NATURAL JOIN companies
```



company_id	item_id	item_name	item_unit	company_name	company_city
16	1	Chex Mix	Pcs	Akas Foods	Delhi
15	2	BN Biscuit	Pcs	Jack Hill Ltd	London
17	3	Mighty Munch	Pcs	Foodies	London
15	4	Pot Rice	Pcs	Jack Hill Ltd	London
18	5	Jaffa Cakes	Pcs	Order All	Boston
15	6	Cheez-It	Pcs	Jack Hill Ltd	London

9.6 SUBQUERIES

Subquery ឬ Inner Query ឬ Nested Query គឺជា query មួយនៅក្នុង query មួយផ្សេងទៀតនិងបង្កប់ក្នុងឃ្លា WHERE ។ Subquery មួយត្រូវបានប្រើដើម្បីបញ្ជូនទិន្នន័យដែលនឹងត្រូវបានប្រើនៅក្នុងសំណួរចម្បងដែលជាលក្ខខណ្ឌដើម្បីរឹតបន្តឹងទិន្នន័យដែលត្រូវទាញយកមកវិញ។ Subqueries អាចត្រូវបានប្រើជាមួយ SELECT, INSERT, UPDATE, និង DELETE រួមជាមួយប្រតិបត្តិករដូចជា =, <, >, >=, <=, IN, BETWEEN ។ ល។

Syntax:

```
SELECT    select_list
FROM      table
WHERE     expr operator
```

```
(SELECT    select_list
FROM      table);
```


EXAMPLE: SUBQUERYIES WITH SELETE

Table: Student

StudentID	Name
V001	Abe
V002	Abhay
V003	Acelin
V004	Adelphos

```
SELECT a.studentid, a.name, b.total_marks
FROM student a, marks b
WHERE a.studentid = b.studentid AND b.total_marks >
(SELECT total_marks
FROM marks
WHERE studentid = 'V002');
```

Table: Marks

StudentID	Total_marks
V001	95
V002	80
V003	74
V004	81



studentid	name	total_marks
V001	Abe	95
V004	Adelphos	81

EXAMPLE: SUBQUERIES WITH INSERT

Table: Customers

ID	Name	Age	Address	Salary
1	Ramesh	35	Ahmedabad	2000.0000
2	Khilan	25	Delhi	1500.0000
3	Kaushik	23	Kota	2000.0000
4	Chaitali	25	Mumbai	6500.0000
5	Hardik	27	Bhopal	8500.0000
6	Komal	22	MP	4500.0000
7	Muffy	24	Indore	10000.0000

```
INSERT INTO CUSTOMER_BKP
SELECT * FROM CUSTOMERS
WHERE ID IN (SELECT ID
FROM CUSTOMERS) ;
```

Table: customer_BKP

ID	Name	Age	Address	Salary
NULL	NULL	NULL	NULL	NULL

EXAMPLE: SUBQUERIES WITH UPDATE

Table: Customers

ID	Name	Age	Address	Salary
1	Ramesh	35	Ahmedabad	2000.0000
2	Khilan	25	Delhi	1500.0000
3	Kaushik	23	Kota	2000.0000
4	Chaitali	25	Mumbai	6500.0000
5	Hardik	27	Bhopal	8500.0000
6	Komal	22	MP	4500.0000
7	Muffy	24	Indore	10000.0000

```
UPDATE CUSTOMERS
SET SALARY = SALARY * 0.25
WHERE AGE IN
(SELECT AGE FROM CUSTOMER_BKP
WHERE AGE >= 27 );
```

EXAMPLE: SUBQUERIES WITH DELETE

Table: Customers

ID	Name	Age	Address	Salary
1	Ramesh	35	Ahmedabad	2000.0000
2	Khilan	25	Delhi	1500.0000
3	Kaushik	23	Kota	2000.0000
4	Chaitali	25	Mumbai	6500.0000
5	Hardik	27	Bhopal	8500.0000
6	Komal	22	MP	4500.0000
7	Muffy	24	Indore	10000.0000

```
DELETE FROM CUSTOMERS
WHERE AGE IN (SELECT AGE FROM CUSTOMERS_BKP
WHERE AGE >= 27 );
```

9.7 CORRELATED SUBQUERIES

- ALL: Used to retrieve records from the main query that match all of the records in the subquery.
- ANY: Used to retrieve records from the main query that match any of the records in the subquery.
- EXISTS: Used to check for the existence of a value in the subquery.
- IN: Used to compare values in a column against column values in another table or query.
- NOT: Used to match any condition opposite of the one defined.
- SOME: Used to retrieve records from the main query that match any of the records in the subquery.

9.7.1 ALL

Syntax:

SELECT *column_name(s)*

FROM *table_name*

WHERE *column_name operator ALL*

(SELECT *column_name* FROM *table_name* WHERE *condition*);

Table: Product

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35

EXAMPLE:

Suppose you want to query the Products table to retrieve product information on products that have less than 20 items in stock. Look at the following script:

```
SELECT ProductID, ProductName, InStock, OnOrder  
FROM Products
```

```
WHERE InStock < ALL (SELECT InStock FROM Products WHERE InStock = 20)
```

Result:

ProductID	ProductName	InStock	OnOrder
AN200	Animated Picture	10	20
BN200	Animated Rainbow	10	20
CE300	Miniature Train Set	1	30
LF300	Friendly Lion	0	30
OT100	Dancing Bird	10	20
RX300	Praying Statue	3	40

9.7.2 ANY

Suppose you want to query the Products table in and the Transactions table to display product information on products that have a productID greater than any productID sold on February 6, 2007. Look at the following script:

```
SELECT *
```

```
FROM Products
```

```
WHERE ProductID > ANY (SELECT ProductID FROM Transactions
```

```
WHERE DateSold = #2/6/07#)
```

Result:

ProductID	ProductName	Price	SalePrice	InStock	OnOrder
LF300	Friendly Lion	\$14.00	\$12.60	0	30
OT100	Dancing Bird	\$10.00	\$9.00	10	20
PO200	Glass Rabbit	\$50.00	\$45.00	50	20
RX300	Praying Statue	\$25.00	\$22.50	3	40
TH100	Crystal Cat	\$75.00	\$67.50	60	20
VR300	China Doll	\$20.00	\$13.00	100	0

9.7.3 EXISTS

Suppose you want to query the Customers table and the Transactions table to retrieve productIDs and dates for products purchased by customers who live in Florida. Look at the following script:

```
SELECT ProductID, DateSold
```

```
FROM Transactions
```

```
WHERE EXISTS (SELECT CustomerID FROM Customers
```

```
WHERE Customers.CustomerID = Transactions.CustomerID
```

```
AND State = 'FL')
```

Result:

ProductID	DateSold
BN200	3/3/2008
CT200	2/5/2008
ET100	2/6/2007
TH100	2/8/2008
CT200	2/22/2008
ET100	2/20/2008
LF300	2/18/2008

9.7.4 IN

Suppose you want to query the Customers table and the Sales table in to retrieve customers who purchased product ID CT200 or productID PO200. Look at the following script:

```
SELECT CustomerID, Lastname, Firstname FROM Customers  
WHERE CustomerID IN  
(SELECT CustomerID FROM Sales  
WHERE ProductID = 'CT200' OR ProductID = 'PO200')
```

Result:

CustomerID ▼	Lastname ▼	Firstname ▼
1	Allison	Kayla
2	Fields	Devin
3	Spencer	Gene

9.8 NESTED SUBQUERIES

Suppose you want to query the Customers table and the Sales table to retrieve the customerID and date of each customer's first purchase.

```
SELECT CustomerID, (SELECT MIN (DateSold) FROM Sales  
WHERE Sales.CustomerID = Customers.CustomerID) AS DateOfFirstPurchase  
FROM Customers  
ORDER BY CustomerID
```

Result:

CustomerID ▼	DateOfFirstPurchase ▼
1	2/17/2008
2	2/22/2008
3	2/5/2008
4	2/10/2007
5	2/6/2007
6	2/18/2008