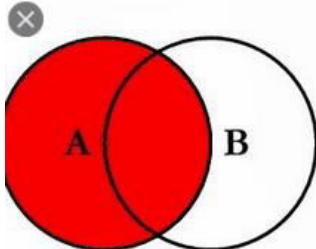


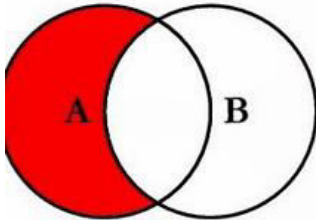
- [Relational model]: Properties*
- [Relational model data integrity constraints]: List
 - [Domain constraint]: Explain*? EG
 - [Entity integrity constraint]: Explain? EG
 - [Referential Integrity constraint]: Explain*? EG (Refer to lecture 6 on how to implement. Know when to use)
 - [Enterprise constraint]: Explain*? EG
- [Relation algebra operators]: Components
 - [Relation specific operators]: List
 - [Restrict]: Explain*? Format*
 - [Project]: Explain*? Format*? EG
 - [Natural Join]: Explain**? Format**? EG
 - [Outer Join]: Explain**? EG**
 - [Division]: Explain? (Keyword ... ALL ...) Not union compatible
 - [Traditional set operators]: Explain (Union compatibility → Positional correspondence + same No. attributes)? List
 - [Union]: Explain? Format? EG
 - [Intersect]: Explain? Format? EG
 - [Difference]: Explain? Format? EG
- [SQL features]: Explain *

10 cards x 2 Min = MAX 20 MINS

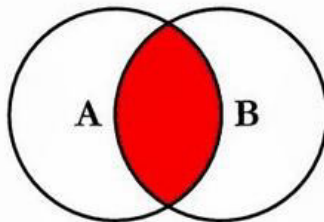
SQL JOINS



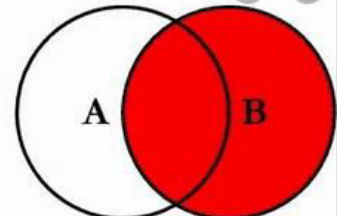
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



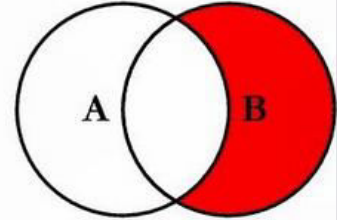
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



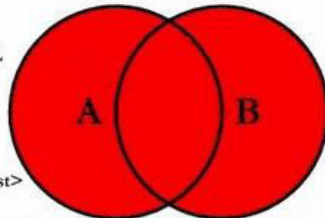
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



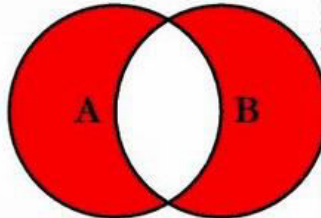
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

Guide for relational algebra

1. Italics all the attributes in the schema
2. Do restrict first
3. Do join the restrict (May not)
4. Do Join all the other relations
5. Do project last

CUSTOMER (**CustID**, *FirstName*, *LastName*, City, Phone, Email)

INVOICE (**InvoiceNumber**, CustID, *Date*)

INVOICE_ITEM(**InvoiceNumber**, **ItemNumber**, Quantity)

ITEM (**ItemNumber**, *ItemName*, UnitPrice)

- a. List the first and last names of Customers who come from the City named Perth.

RESTRICT City = 'Perth' (CUSTOMER) → Relation1

PROJECT FirstName, LastName (CUSTOMER) → Solution

- b. List the first and last names of customers who had transactions on 1st August 2020.

RESTRICT Date = '01/08/2020' (Invoice) → Relation1

Relation1* Relation1.CustID = CUSTOMER.CustID CUSTOMER → Relation2

PROJECT FirstName, LastName (Relation2) → Solution

- c. List the price of the item called "Back Scratcher"

RESTRICT ItemName = 'Back Scratchers' (ITEM) → T1

PROJECT UnitPrice (T1) → Solution

- d. List the first and last names of any customer who has purchased more than 10 "Back Scratchers" in a single transaction.

RESTRICT Quantity = 10 (INVOICE_ITEM) → Relation1

RESTRICT ItemName = 'Back Scratchers'(ITEM) → Relation2

Relation1* Relation1.ItemNumber = Relation2.ItemNumber Relation2 → Relation3

Relation3* Relation3.InvoiceNumber = INVOICE.InvoiceNumber INVOICE → Relation4

Relation4* Relation4.CustID = CUSTOMER.CustID CUSTOMER → FinalRelation

PROJECT FirstName, LastName (FinalRelation) → SOLUTION

- e. List the names and quantities of items purchased on 1st August 2020 by the customer Peter Simpson.

RESTRICT Date = '01/08/2020' (Invoice) → Relation1
RESTRICT FirstName = 'Peter' AND LastName = 'Simpson' (CUSTOMER) → Relation2
Relation2* Relation2.CustID = Relation3.CustID Relation3 → Relation3

INVOICE_ITEM* INVOICE_ITEM.InvoiceNumber = Relation3.InvoiceNumber Relation3 → T4

T4* T4.ItemNumber = ITEM.ItemNumber ITEM → T5

PROJECT ItemName, Quantity (T5) → Solution

f. List the dates on which Homer Griffin made purchases.

g. List the first and last names of customers who have bought "Back Scratcher" or "Hair Remover"

RESTRICT ItemName = 'Back Scratcher' (ITEM) → T1
INVOICE_ITEM* INVOICE_ITEM.ItemNumber = T1.ItemNumber T1 → T2
T2* T2.InvoiceNumber = INVOICE.InvoiceNumber INVOICE → T3
T3* T3.CustID = CUSTOMER.CustID CUSTOMER → T4

PROJECT FirstName, LastName (T4) → Final1 ****Union compatibility specifically positional correspondence + same No. of Attributes****

RESTRICT ItemName = 'Back Scratcher' (ITEM) → S1
INVOICE_ITEM* INVOICE_ITEM.ItemNumber = S1.ItemNumber S1 → S2
S2* S2.InvoiceNumber = INVOICE.InvoiceNumber INVOICE → S3
S3* S3.CustID = CUSTOMER.CustID CUSTOMER → S4

PROJECT FirstName, LastName (S4) → Final2 ****Union compatibility specifically positional correspondence + same No. of Attributes****

Final1 UNION Final2 → Solution

h. List the first and last names of customers who have bought "Back Scratcher" **but have not** bought "Hair Remover"

RESTRICT ItemName = 'Back Scratcher' (ITEM) → T1

INVOICE_ITEM* INVOICE_ITEM.ItemNumber = T1.ItemNumber T1 → T2

T2* T2.InvoiceNumber = INVOICE.InvoiceNumber INVOICE → T3

T3* T3.CustID = CUSTOMER.CustID CUSTOMER → T4

PROJECT FirstName, LastName (T4) → Final1 ****Union compatibility specifically positional correspondence + same No. of Attributes****

RESTRICT ItemName = 'Hair Remover' (ITEM) → T5

INVOICE_ITEM* INVOICE_ITEM.ItemNumber = T5.ItemNumber T5 → T6

T6* T6.InvoiceNumber = INVOICE.InvoiceNumber INVOICE → T7

T8* T8.CustID = CUSTOMER.CustID CUSTOMER → T8

PROJECT FirstName, LastName (T8) → Final2 ****Union compatibility specifically positional correspondence + same No. of Attributes****

Final1 **MINUS** Final2 → Solution

- i. List the first and last names of customers who have bought “Back Scratcher” **and** “Hair Remover”

RESTRICT ItemName = 'Back Scratcher' (ITEM) → T1

INVOICE_ITEM* INVOICE_ITEM.ItemNumber = T1.ItemNumber T1 → T2

T2* T2.InvoiceNumber = INVOICE.InvoiceNumber INVOICE → T3

T3* T3.CustID = CUSTOMER.CustID CUSTOMER → T4

PROJECT FirstName, LastName (T4) → Final1 ****Union compatibility specifically positional correspondence + same No. of Attributes****

RESTRICT ItemName = 'Hair Remover' (ITEM) → T5

INVOICE_ITEM* INVOICE_ITEM.ItemNumber = T5.ItemNumber T5 → T6

T6* T6.InvoiceNumber = INVOICE.InvoiceNumber INVOICE → T7

T7* T7.CustID = CUSTOMER.CustID CUSTOMER → T8

PROJECT FirstName, LastName (T8) → Final2 ****Union compatibility specifically positional correspondence + same No. of Attributes****

Final1 **INTERSECTION** Final2 → Solution

- j. List the first and last names of any customers who have bought **all** of the items listed in the Item relation. (This does not need to be as part of a single purchase).

INVOICE* INVOICE.CustID = CUSTOMER.CustID CUSTOMER → T1

T1 LEFT OUTER JOIN T1.InvoiceNumber = INVOICE_ITEM.InvoiceNumber INVOICE_ITEM → T2

ITEM LEFT OUTER JOIN ITEM.ItemNumber = T2.InvoiceNumber T2 → T3

PROJECT FirstName, LastName, ItemNumber (T3) → Final1

PROJECT ItemNumber (ITEM) → Final2

Final1 **DIVIDEBY** Final2 → Solution