Joins and Unions

CMPE 2400

Databases

Introduction

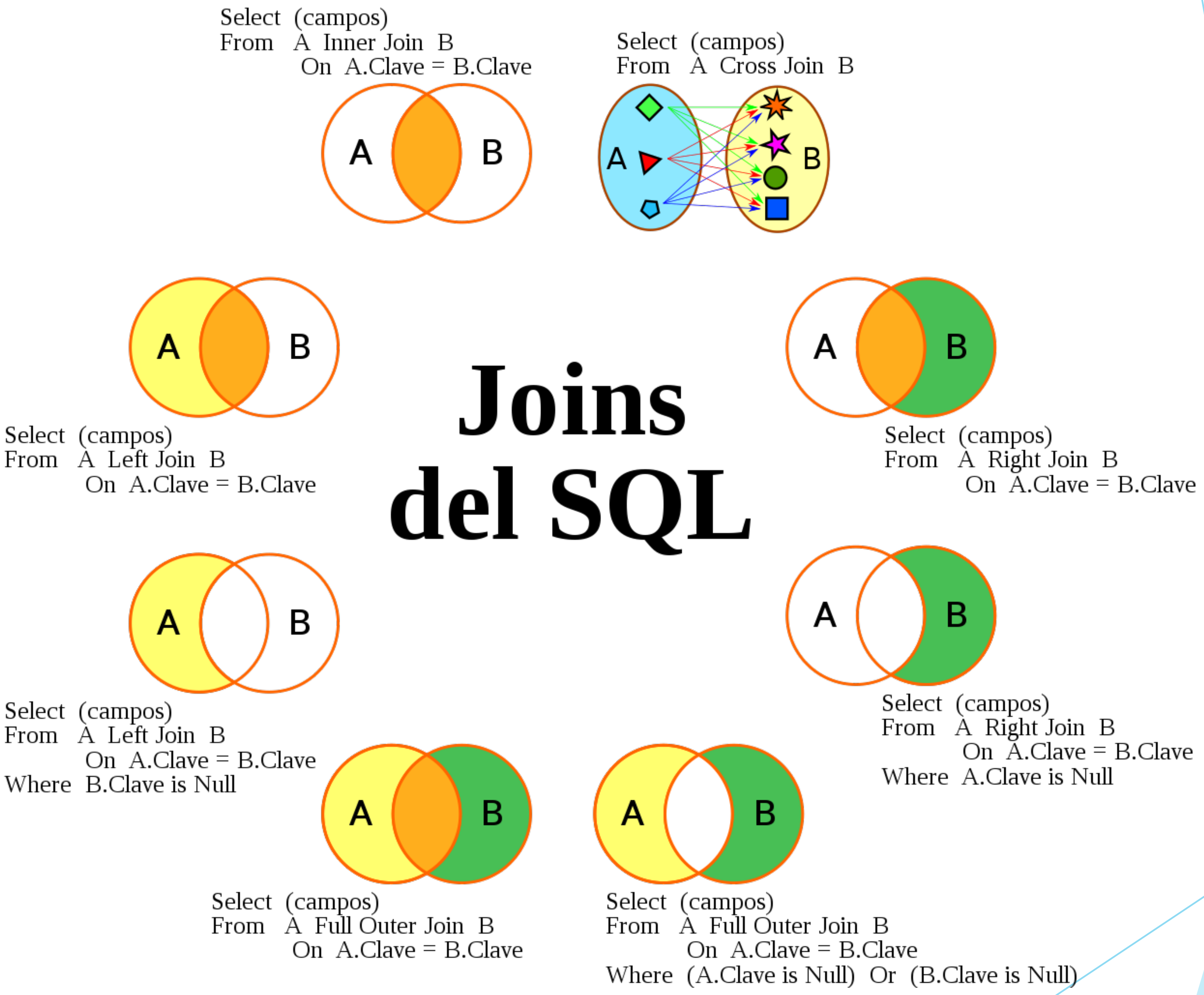
When data is *normalized*, it is stored in relation to other data

We often want to see the data combined, for reporting or other uses

Time for JOINS!

Join Types:

- INNER **JOIN**
- **CROSS JOIN**
- **LEFT** OUTER **JOIN**
- **RIGHT** OUTER **JOIN**
- **FULL** OUTER **JOIN**



Joins del SQL

```
Select  (campos)
From   A  Inner Join  B
          On  A.Clave = B.Clave
```

```
Select  (campos)
From   A  Cross Join  B
```

```
Select  (campos)
From   A  Left Join  B
          On  A.Clave = B.Clave
```

```
Select  (campos)
From   A  Right Join  B
          On  A.Clave = B.Clave
```

```
Select  (campos)
From   A  Left Join  B
          On  A.Clave = B.Clave
Where  B.Clave is Null
```

```
Select  (campos)
From   A  Right Join  B
          On  A.Clave = B.Clave
Where  A.Clave is Null
```

```
Select  (campos)
From   A  Full Outer Join  B
          On  A.Clave = B.Clave
```

```
Select  (campos)
From   A  Full Outer Join  B
          On  A.Clave = B.Clave
Where  (A.Clave is Null)  Or  (B.Clave is Null)
```

Source: https://www.dreaminterview.com/sql/joins-in-sql/

inner join

- Inner joins relate 2 tables using a common column, or combination of several common columns.
  - The returned data set will contain the rows where the values from the compared column(s) are equal.
  - Typically the relationship is a foreign key to primary key relationship, but any column(s) of equal data types may be used.
  - If there is a 1 to N relationship, this can yield multiple rows for a single chosen row in the ‘1’ table.

Syntax

- ANSI Syntax
  - Universal base standard that may be used across DBMS platforms
- SQL Server Syntax
  - Proprietary to Microsoft
  - More straight-forward for table designation syntax, but result in more complicated conditional (filter) statements

Syntax

- In both syntax forms, a table name must be specified with a column name if the selected column exists in both tables.
  - Failure to follow this protocol will result in an ambiguous command to the SQL server.
  - The server will quite happily accost you with an error message to inform you of your blunder.

ANSI inner join Syntax

```
select <[TableName.]ColumnName> [as 'ColumnAlias'],
       <[TableName.]ColumnName> [as 'ColumnAlias']
from   <Table1Name> [as TableAlias]
   inner join  <Table2Name> [as TableAlias]
   on          <JoinCondition(s)>
where   [<SearchCondition(s)>]
order by    [<OrderingCondition(s)>]
```

ANSI Syntax without table alias

```
select  stores.stor_name  as 'Store',
    stores.stor_id        as 'Store ID',
    sales.title_id        as 'Title ID',
    sales.payterms        as 'Payment Terms',
    sales.ord_date        as 'Order Date'
from       PublishersDatabase.dbo.stores
 inner join    PublishersDatabase.dbo.sales
 on     stores.stor_id = sales.stor_id
where   sales.payterms = 'Net 60'
order by   stores.stor_name desc
```

```
select  st.stor_name  as 'Store',
        st.stor_id    as 'Store ID',
        s.title_id    as 'Title ID',
        s.payterms    as 'Payment Terms',
        s.ord_date    as 'Order Date'
from       PublishersDatabase.dbo.stores as st
 inner join    PublishersDatabase.dbo.sales  as s
 on     st.stor_id = s.stor_id
where   s.payterms = 'Net 60'
order by   st.stor_name desc
```

ANSI Syntax with table alias

SQL Server inner join Syntax

```
select  <[TableName.]ColumnName> ['ColumnAlias'],
        <[TableName.]ColumnName> ['ColumnAlias']
from    <Table1Name> [Table1Alias],
        <Table2Name> [Table2Alias]
where   <Joining SearchCondition(s)>
and     <OtherCriteriaConditions>
order by   <OrderingCondition(s)>
```

```
select stores.stor_name     'Store',
       stores.stor_id       'Store ID',
       sales.title_id       'Title ID',
       sales.payterms       'Payment Terms',
       sales.ord_date       'Order Date'
from   PublishersDatabase.dbo.stores,
       PublishersDatabase.dbo.sales
where  stores.stor_id = sales.stor_id
  and  sales.payterms = 'Net 60'
order by   stores.stor_name desc
```

SQL Server Syntax without table alias

SQL Server Syntax

```
select stores.stor_name     'Store',
       stores.stor_id       'Store ID',
       sales.title_id       'Title ID',
       sales.payterms       'Payment Terms',
       sales.ord_date       'Order Date'
from   PublishersDatabase.dbo.stores,
       PublishersDatabase.dbo.sales
where  stores.stor_id = sales.stor_id
  and  sales.payterms = 'Net 60'
order by   stores.stor_name desc
```

1

Using the Publishers schema, display a list of all male employee names and their job descriptions

- Use Aliases for all columns and tables
- Format name as LastName, FirstName
- Use descending order by job level for your result set

2

Write your solutions for ANSI and MS-SQL notation

- Use explicit notation for the ANSI solution
- Use implicit notation for the SQL solution

Exercise – inner join

Exercise Solution

- ANSI notation

```
select e.lname + ', `
       + e.fname as 'Employee Name',
       j.job_desc as 'Job Description'
from   PublishersDatabase.dbo.employee e
   inner join PublishersDatabase.dbo.jobs j
   on  e.job_id = j.job_id
where  right(e.emp_id, 1) = 'M'
order by  e.job_lvl desc
```

Exercise Solution

- MS-SQL notation

```
select  lname + ', `
     + fname 'Employee Name',
     job_desc 'Job Description'
from
 PublishersDatabase.dbo.employee e,
 PublishersDatabase.dbo.jobs j
where   e.job_id = j.job_id
 and    right(emp_id, 1) = 'M'
order by   job_lvl desc
```

Combining Joins

Joins may be combined within the same select statement.

There is no limitation as to what kind of joins may be combined. We will begin by combining inner joins only.

Combined joins are commonly needed to bring together data with an N-to-M relationship.

Remember that the N-to-M relationship is accomplished via an intermediate linking table.

An example from PUBS

```sql
select        stor_name     as 'Store',
              st.stor_id    as 'Store ID',
              title         as 'Book Title',
              [type]        as 'Book Category',
              payterms      as 'Payment Terms',
              ord_date      as 'Order Date'
from          PublishersDatabase.dbo.stores as st
    inner join     PublishersDatabase.dbo.sales  as s
         on        st.stor_id = s.stor_id
     inner join    PublishersDatabase.dbo.titles as t
         on        s.title_id = t.title_id
where         payterms = 'Net 60'
order by stor_name desc
```

Exercise

- Print the author names, author city and book title for all books where the author was advanced more than $4000.
  - Use Aliases for all columns and tables
  - Format name as **LastName, FirstName**
  - Order results by the city the author lives in.

Solution

```
select   au_lname + ', `
         + au_fname  as 'Author''s Name',
         city        as 'Author City',
         title       as 'Book Title'
from     PublishersDatabase.dbo.authors as a
inner join PublishersDatabase.titleauthor ta
 on      a.au_id = ta.au_id
 inner join  PublishersDatabase.dbo.titles as t
  on     ta.title_id = t.title_id
where    advance > 4000
order by     city
```