Veri Tabanı Yönetimi ve Modellemesi

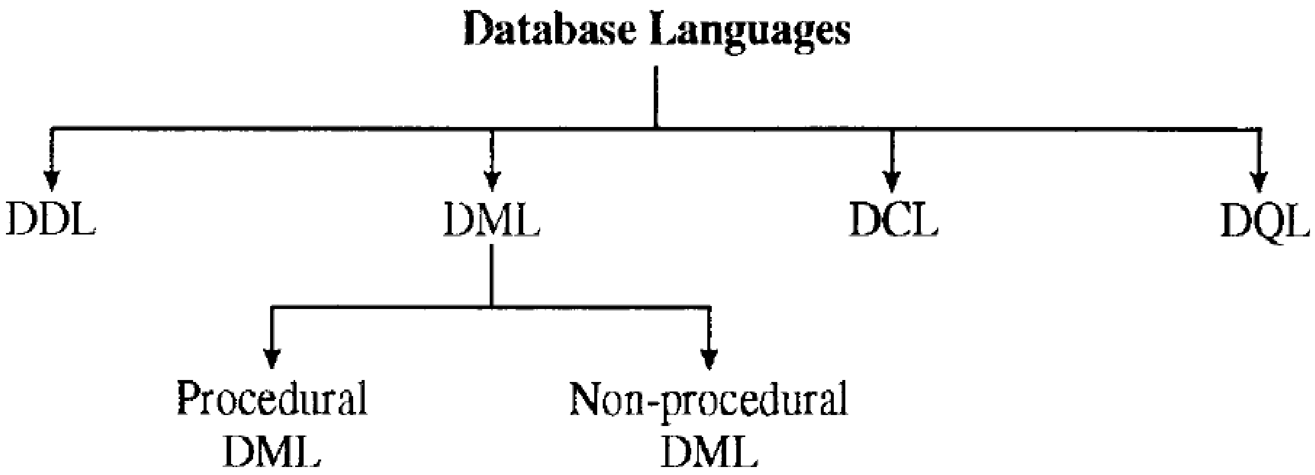
HAFTA 5

Haftalık Ders Akışı

1. Veritabanı Kavramlarına Giriş
2. Veri Tabanı Türleri, İlişkisel Veri Tabanı Tasarımı
3. ER Diyagramları ve Normalizasyon
4. SQL Server Arayüzü, Veri Tabanı Nesneleri
5. **T-SQL ve SQL Sorguları**
6. İndeks ve View
7. Stored Procedure ve Fonksiyonlar
8. Ara Sınav
9. Tetikleyiciler
10. Transaction Kavramları ve Yedekleme
11. Kullanıcı Türleri ve Kullanıcı Yönetimi
12. No-SQL Veri Tabanları
13. No-SQL Veri Tabanları
14. Proje Sunumu
15. Proje Sunumları

Veritabanı Dilleri

- Data Definition Language (DDL)
  - Create
  - Alter
  - Drop
  - Truncate
  - Comment
- Data Manipulation Language (DML)
  - Select
  - Insert
  - Update
  - Delete
  - Lock Table
- Data Control Language (DCL)
  - Commit
  - Roll-Back
  - Save Point
  - Grant/Revoke
  - Set Transaction
- Data Ouery Language (DQL)



CREATE

- Database
- Schema
- Table
- İndex
- View
- UDF
- Store Procedure
- Trigger
- Users

Create

- create database Db_Name

- CREATE DATABASE [Zoo] ON PRIMARY

```
( NAME = N'Zoo', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\Zoo.mdf' , SIZE = 3072KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )

 LOG ON

( NAME = N'Zoo_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\Zoo_log.ldf' , SIZE = 1024KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)

GO
```

Create

```
CREATE TABLE Table_Name
(columnName  valueType,
columnName valueType,
columnName valueType NOT NULL,...
PRIMARY KEY (columnName))
```

Create

```
CREATE TABLE [dbo].[tbl_Staf](
        [id] [int] NOT NULL,
        [name] [varchar](50) NULL,
        [surname] [varchar](50) NULL,
        [birthDate] [datetime] NULL,
        [StartDate] [datetime] NULL,
        [stafRole] [int] NULL,
 CONSTRAINT [PK_tbl_Staf] PRIMARY KEY CLUSTERED( [id] ASC )
WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON) ON [PRIMARY] ) ON [PRIMARY]
ALTER TABLE [dbo].[tbl_Staf]  WITH CHECK ADD  CONSTRAINT [FK_tbl_Staf_tbl_Staf] FOREIGN
KEY([stafRole])  REFERENCES [dbo].[tbl_StafRole] ([id])
ALTER TABLE [dbo].[tbl_Staf] CHECK CONSTRAINT [FK_tbl_Staf_tbl_Staf]
```

ALTER

- ALTER TABLE Table_Name ADD (COLUMN) column_Name dataType;
- ALTER TABLE Table_Name DROP COLUMN column_Name
- ALTER TABLE [dbo].[tbl_AnimalType] WITH CHECK ADD CONSTRAINT [FK_tbl_AnimalType_tbl_AnimalType] FOREIGN KEY([ClassTypeId])

  REFERENCES [dbo].[tbl_AnimalType] ([id])

DROP

- DROP DATABASE db_Name
- DROP table tbl_Name
- ALTER TABLE Table_Name DROP COLUMN column_Name
- USE [Zoo]

  ```
  GO

  IF  EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[dbo].[FK_tbl_AnimalType_tbl_AnimalType]') AND parent_object_id = OBJECT_ID(N'[dbo].[tbl_AnimalType]'))

  ALTER TABLE [dbo].[tbl_AnimalType] DROP CONSTRAINT [FK_tbl_AnimalType_tbl_AnimalType]

  IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[tbl_AnimalType]') AND type in (N'U'))

  DROP TABLE [dbo].[tbl_AnimalType]
  ```

TRUNCATE

- Truncate table tbl_Name
  - Truncate table tbl_AnimalType

SELECT

- Select * from tbl_Name
  - Food Type tablosundaki verileri getiren sorgu:
    - Select * from tbl_FoodType
- Select columnName,... From tbl_Name
  - Food Type tablosundaki id ve tanım bilgilerini getiren sorgu:
    - Select id, description FROM tbl_FoodType

Top

- Select Top 10 * from tbl_Name
    - Food Type tablosundaki ilk 10 yemek türünü getiren soru:
        - Select top 10* from tbl_FoodType

Where

- Select * from tbl_Name Where Statemen(s)
  - Food Type tablosunda Ana Kategorileri getiren sorgu:
    - Select * FROM tbl_FoodType Where ParentId is NULL
  - Food Type tablosunda 2. kategoriye bağlı türleri getiren sorgu:
    - Select * from tbl_FoodType Where ParentId=2
  - 2019 yılından bugüne hayvanat bahçesine gelen hayvaları getiren sorgu:
    - select * from tbl_Animal where arrivalDate>='01.01.2019'

Koşul İfadeleri

- =, >,<,>=,<=, <>, !=, LIKE

- Food Type tablosunda 2. kategoriye bağlı türleri getiren sorgu:
  - Select * from tbl_FoodType Where ParentId=2
- 2019 yılından bugüne hayvanat bahçesine gelen hayvaları getiren sorgu:
  - Select * from tbl_FoodType Where arrivalDate>='01.01.2019‘
- Food Type tablosunda 2. Kategorinin alt ürünleri dışındaki ürünler:
  - Select * from tbl_FoodType Where ParentId!=2
  - Select * from tbl_FoodType Where ParentId<>2
- İsminin içinde Et geçen yemek türleri
  - select * from tbl_FoodType where description Like 'et‘
  - select * from tbl_FoodType where description Like '%et‘
  - select * from tbl_FoodType where description Like 'et%‘
  - select * from tbl_FoodType where description Like '%et%'

AND, OR, NOT

- SELECT columnName, ... FROM tbl_name WHERE condition1 AND condition2 AND condition3 ...;
  - Geliş tarihi 01.01.2019 dan büyük eşit ve Hayvan türü 3 olan hayvanlar:
    - select * from tbl_Animal where arrivalDate>='01.01.2019‘ and animalType_Id=3

- SELECT columnName, ... FROM tbl_name WHERE condition1 OR condition2
  - Doğum tarihi yada geliş tarihi 01.01.2019 dan itibaren hayvanat bahçesine katılan hayvanlar:
    - select * from tbl_Animal where arrivalDate>='01.01.2019' or birthDate>='01.01.2019'

- SELECT columnName, ... FROM tbl_name WHERE NOT Condition
  - Food Type tablosundaki alt yemek türleri
    - Select * FROM tbl_FoodType Where ParentId is NOT NULL

DISTINCT

- Select distinct columnName from tbl_Name
  - Hayvanat bahçesinde yaşayan hayvan türleri
    - select distinct animalType_Id from tbl_Animal

ORDER BY

- SELECT columnName, ... FROM tbl_name ORDER BY column1, column2, ... ASC|DESC
  - Doğum tarihi yada geliş tarihi 01.01.2019 dan itibaren hayvanat bahçesine katılan hayvanların türlerine göre sıralı gösterimi:
    - select * from tbl_Animal where (arrivalDate>='01.01.2019' or birthDate>='01.01.2019') order by animalType_Id
  - Doğum tarihi yada geliş tarihi 01.01.2019 dan itibaren hayvanat bahçesine katılan hayvanların türlerine göre artan sıralı gösterimi
    - select * from tbl_Animal where (arrivalDate>='01.01.2019' or birthDate>='01.01.2019') order by animalType_Id,id asc
  - Doğum tarihi yada geliş tarihi 01.01.2019 dan itibaren hayvanat bahçesine katılan hayvanların türlerine göre azalan sıralı gösterimi
    - select * from tbl_Animal where (arrivalDate>='01.01.2019' or birthDate>='01.01.2019') order by animalType_Id desc

GROUP BY

- SELECT columnName,... FROM tbl_name WHERE condition GROUP BY column_name(s) ORDER BY column_name(s);
  - Hayvanat bahçesinde hangi hayvan türünden kaç adet bulunmaktadır?
  - select animalType_Id, COUNT(id) AS NumberOfTypes from tbl_Animal group by animalType_Id order by animalType_Id

T-Sql ifadeleri

- ABS: Mutlak değer
  - select ABS(-3)
- CEILING/FLOOR/ROUND: Yuvarlama işlemi
  - select CEILING(1.34)
  - select FLOOR(1.34)
  - select ROUND(1.34,1)
- RAND: [0,1] arasında rastgele değer üretir
  - select RAND()
- SQRT: Karekök alma
  - select SQRT(4)

T-Sql İfadeleri

- Count
  - select COUNT(id) from tbl_FoodType where parentId is null
- Max
  - select MAX(id) from tbl_Animal
- Min
  - select min(id) from tbl_Animal

T-Sql İfadeleri

- CHAR
  - select CHAR(65)
- CHARINDEX
  - select CHARINDEX('l','ali veli',1)
- LEFT
  - Select LEFT('ali veli',3)
- RIGHT
  - select RIGHT('ali veli',2)
- LEN
  - select LEN('ali veli')
- UPPER
  - select UPPER('ali veli‘)
- LOWER
  - select LOWER('ALİ VELİ')
- LTRIM
  - select LTRIM('   ali veli')
- RTRIM
  - select RTRIM('ali veli   ')
- REPLACE
  - select REPLACE('ali veli','veli',‘yılmaz')
- REVERSE
  - select REVERSE('ali')
- SUBSTRING
  - Select SUBSTRING('ali veli',3,2)

T-Sql İfadeleri

- GETDATE()
  - select GETDATE()
  - select CONVERT(varchar,GETDATE(),105)
- DATEPART
  - select DATEPART(wk,getdate())/ select DATEPART(dy,getdate()) / select DATEPART(m,getdate())
- DAY/MONTH/YEAR
  - select DAY(getdate()) /select DAY(Convert(datetime,'18.12.2018',104))
  - select MONTH(GETDATE())
  - select YEAR(GETDATE())
- DATEADD
  - select DATEADD(day,3,GETDATE())
  - select DATEADD(MONTH,3,GETDATE())
  - select DATEADD(YEAR,3,GETDATE())

T-Sql İfadeleri

- DATEDIFF
  - select DATEDIFF(day,Convert(datetime,'16.10.2019',104),convert(datetime,'1.11.2019',104))
  - select DATEDIFF(MONTH,Convert(datetime,'16.10.2019',104),convert(datetime,'1.11.2019',104))
  - select DATEDIFF(year,Convert(datetime,'16.10.2019',104),convert(datetime,'1.11.2019',104))
- DATENAME
  - SELECT DATENAME(M, '2017/08/25') AS DatePartString;

IN - NOT IN

- IN
  - Papağan, Karga ve penguen türündeki hayvanların bilgileri:
    - select * from tbl_AnimalType where info in ('Papağan','Karga','penguen')

- NOT IN
  - Papağan, Karga ve penguen türleri dışındaki hayvanların bilgileri:
    - select * from tbl_AnimalType where info not in ('Papağan','Karga','penguen')

Çoklu Sorgular

- IN
  - select * from tbl_AnimalType where id in (select animalType_Id from tbl_Animal)

- EXISTS
  - Hayvanat bahçesindeki mevcut olan hayvan türleri:
  - select * from tbl_AnimalType where EXISTS(select animalType_Id from tbl_Animal where animalType_Id=tbl_AnimalType.id)

- NOT EXISTS
  - Hayvanat bahçesinde olmayan hayvan türleri:
  - select * from tbl_AnimalType where NOT EXISTS(select animalType_Id from tbl_Animal where animalType_Id=tbl_AnimalType.id)

Çoklu Sorgular

- AS
- Hayvan bilgilerinin yanından türlerinin de isimleri yazılması:
  - select *, (select info from tbl_AnimalType where id=animalType_Id)AS TypeName from tbl_Animal

Çoklu Sorgular

Klasik Join

Inner Join

Outer Join

Cross Join

Klasik Join

```
select * from tbl_AnimalType where id in (select animalType_Id from tbl_Animal)
```

INNER JOIN

- Select * from tbl_Name1 [INNER] JOIN tbl_Name2

  ON tbl_Name1 .ColumnName= tbl_Name2.ColumnName

  - select * from tbl_Animal inner join tbl_animalType

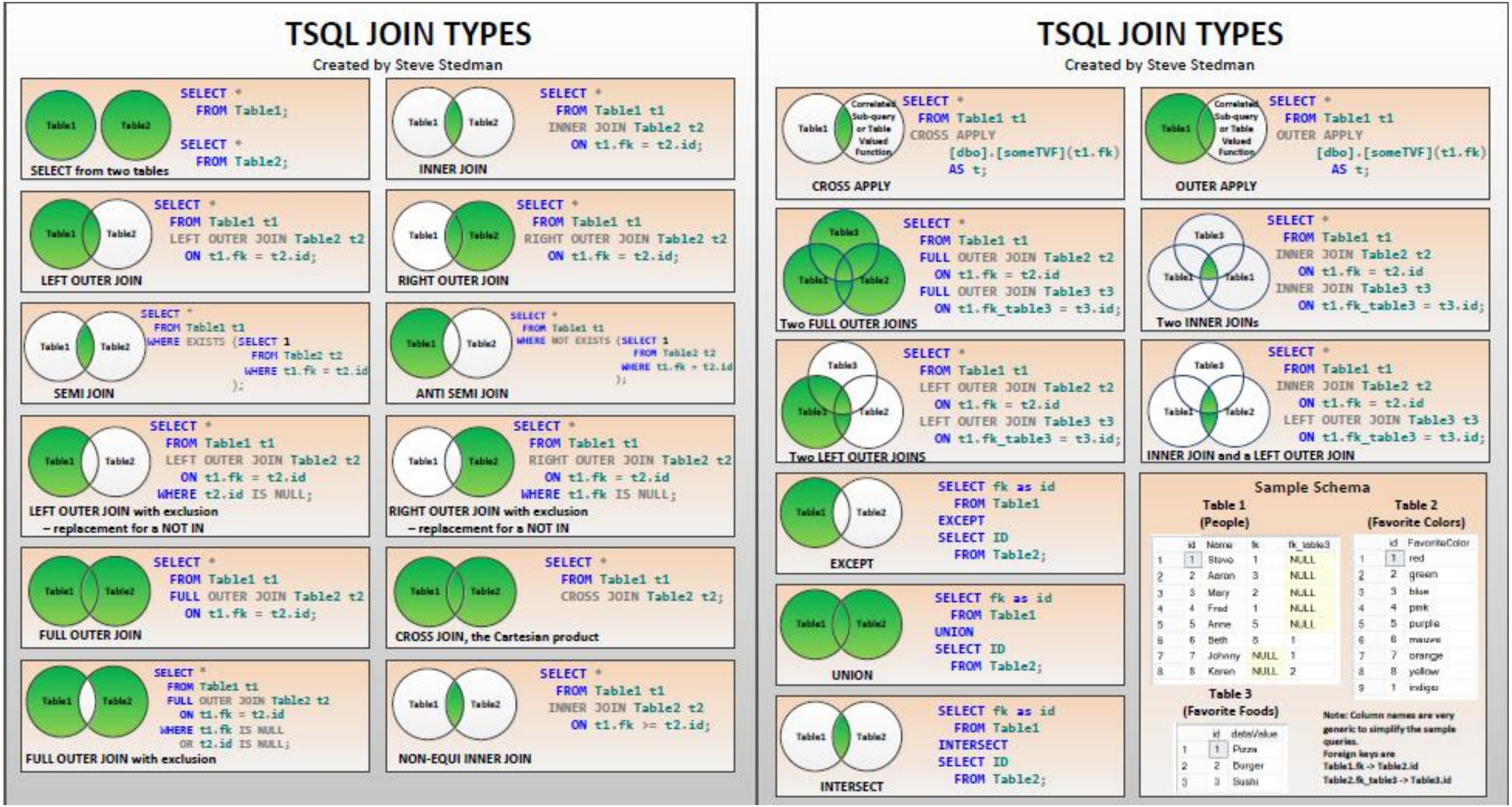
    on tbl_animal.animalType_Id =tbl_AnimalType.id

OUTER JOIN

- LEFT
  - select * from tbl_Animal LEFT join tbl_animalType
  on tbl_animal.animalType_Id =tbl_AnimalType.id
- RIGHT
  - select * from tbl_Animal RIGHT join tbl_animalType
  on tbl_animal.animalType_Id =tbl_AnimalType.id
- FULL
  - select * from tbl_Animal FULL join tbl_animalType
  on tbl_animal.animalType_Id =tbl_AnimalType.id

CROSS JOIN

- SELECT * from tbl_Name1 CROSS JOIN tbl_Name2
  - select * from tbl_Animal cross join tbl_animalType



TSQL JOIN TYPES

Created by Steve Stedman

```
SELECT *
  FROM Table1;

SELECT *
  FROM Table2;
```

SELECT from two tables

```
SELECT *
  FROM Table1 t1
 INNER JOIN Table2 t2
    ON t1.fk = t2.id;
```

INNER JOIN

```
SELECT *
  FROM Table1 t1
  LEFT OUTER JOIN Table2 t2
    ON t1.fk = t2.id;
```

LEFT OUTER JOIN

```
SELECT *
  FROM Table1 t1
 RIGHT OUTER JOIN Table2 t2
    ON t1.fk = t2.id;
```

RIGHT OUTER JOIN

```
SELECT *
  FROM Table1 t1
 WHERE EXISTS (SELECT 1
                 FROM Table2 t2
                WHERE t1.fk = t2.id
              );
```

SEMI JOIN

```
SELECT *
  FROM Table1 t1
 WHERE NOT EXISTS (SELECT 1
                     FROM Table2 t2
                    WHERE t1.fk = t2.id
                  );
```

ANTI SEMI JOIN

```
SELECT *
  FROM Table1 t1
  LEFT OUTER JOIN Table2 t2
    ON t1.fk = t2.id
 WHERE t2.id IS NULL;
```

LEFT OUTER JOIN with exclusion
– replacement for a NOT IN

```
SELECT *
  FROM Table1 t1
  RIGHT OUTER JOIN Table2 t2
    ON t1.fk = t2.id
 WHERE t1.fk IS NULL;
```

RIGHT OUTER JOIN with exclusion
– replacement for a NOT IN

```
SELECT *
  FROM Table1 t1
  FULL OUTER JOIN Table2 t2
    ON t1.fk = t2.id;
```

FULL OUTER JOIN

```
SELECT *
  FROM Table1 t1
 CROSS JOIN Table2 t2;
```

CROSS JOIN, the Cartesian product

```
SELECT *
  FROM Table1 t1
  FULL OUTER JOIN Table2 t2
    ON t1.fk = t2.id
 WHERE t1.fk IS NULL
    OR t2.id IS NULL;
```

FULL OUTER JOIN with exclusion

```
SELECT *
  FROM Table1 t1
 INNER JOIN Table2 t2
    ON t1.fk >= t2.id;
```

NON-EQUI INNER JOIN

TSQL JOIN TYPES

Created by Steve Stedman

```
SELECT *
  FROM Table1 t1
 CROSS APPLY
       [dbo].[someTVF](t1.fk)
    AS t;
```

CROSS APPLY

```
SELECT *
  FROM Table1 t1
 OUTER APPLY
       [dbo].[someTVF](t1.fk)
    AS t;
```

OUTER APPLY

```
SELECT *
  FROM Table1 t1
  FULL OUTER JOIN Table2 t2
    ON t1.fk = t2.id
  FULL OUTER JOIN Table3 t3
    ON t1.fk_table3 = t3.id;
```

Two FULL OUTER JOINS

```
SELECT *
  FROM Table1 t1
 INNER JOIN Table2 t2
    ON t1.fk = t2.id
 INNER JOIN Table3 t3
    ON t1.fk_table3 = t3.id;
```

Two INNER JOINs

```
SELECT *
  FROM Table1 t1
  LEFT OUTER JOIN Table2 t2
    ON t1.fk = t2.id
  LEFT OUTER JOIN Table3 t3
    ON t1.fk_table3 = t3.id;
```

Two LEFT OUTER JOINS

```
SELECT *
  FROM Table1 t1
 INNER JOIN Table2 t2
    ON t1.fk = t2.id
  LEFT OUTER JOIN Table3 t3
    ON t1.fk_table3 = t3.id;
```

INNER JOIN and a LEFT OUTER JOIN

```
SELECT fk as id
  FROM Table1
EXCEPT
SELECT ID
  FROM Table2;
```

EXCEPT

```
SELECT fk as id
  FROM Table1
UNION
SELECT ID
  FROM Table2;
```

UNION

```
SELECT fk as id
  FROM Table1
INTERSECT
SELECT ID
  FROM Table2;
```

INTERSECT

Sample Schema

Table 1
(People)

	id	Name	fk	fk_table3
1	1	Steve	1	NULL
2	2	Aaron	3	NULL
3	3	Mary	2	NULL
4	4	Fred	1	NULL
5	5	Anne	5	NULL
6	6	Beth	8	1
7	7	Johnny	NULL	1
8	8	Karen	NULL	2

Table 2
(Favorite Colors)

	id	FavoriteColor
1	1	red
2	2	green
3	3	blue
4	4	pink
5	5	purple
6	6	mauve
7	7	orange
8	8	yellow
9	1	indigo

Table 3
(Favorite Foods)

	id	dataValue
1	1	Pizza
2	2	Burger
3	3	Sushi

Note: Column names are very generic to simplify the sample queries.
Foreign keys are
Table1.fk -> Table2.id
Table2.fk_table3 -> Table3.id

UNION

-
  ```
  SELECT * FROM tbl_Name1
              UNION
  SELECT * FROM tbl_Name2
  ```

  -
    ```
    select * from tbl_AnimalType
            union
    select * from tbl_FoodType
    ```

INSERT

- Insert into tbl_name (columnName,...) values (valuesofData,...)
  - insert into tbl_FoodType (id, [description],parentId) values (1,‘Tohum', null)
  - insert into tbl_FoodType ([description],parentId) values (‘Tohum', null)
  - insert into tbl_FoodType (parentId,[description]) values (null,’Tohum')
- Insert into tbl_name values (valuesofData,...)
  - insert into tbl_FoodType values (1,‘Tohum', null)
  - insert into tbl_AnimalType values ('Maymun',null)
- Select columnName,... İNTO new_Tbl from tbl_name
  - select * into tbl_tmpFoodType from tbl_FoodType
  - select id,[description] into tbl_tmpFoodType from tbl_FoodType

UPDATE

- Update tbl_name

  Set columnName=newValue

  Where statement

- update tbl_AnimalType

  set ClassTypeId=16

  where info in('Orangutan',‘Şempaze')

DELETE

- DELETE FROM tblName WHERE statement(s)
  - Delete from tbl_FoodType where [description]=‘Tohum’
  - delete from tbl_AnimalType where info='maymun‘

Proje Önerisi

Eksikliklerin tamamlanması için;

ProjeÖnerisi ve Ödev 1 ikisi birlikte tek döküman olarak yüklenecektir.

Sisteme yüklenen dokümanlar ogrenciNo.pdf, ogrenciNo.doc,... Vb şeklinde olmalıdır.