

DATABASE

Kaydol X

Hızlı ve kolaydır.

Adın Soyadın

Cep telefonu numarası veya e-posta

Yeni şifre

Doğum Tarihi ?
1 Eki 2020

Cinsiyet ?
 Kadın Erkek Özel

Kaydol düğmesine tıklayarak, Koşullarımıza, Veri İlkemizi ve Çerezler İlkemizi kabul etmiş olursun. Bizden SMS Bildirimleri alabilir ve bu bildirimleri istediğiniz zaman durdurabilirsin.

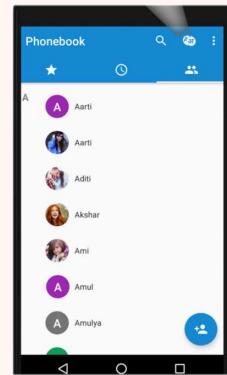
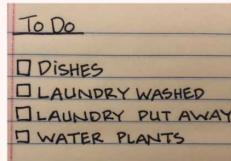
Kaydol

```
public class facebook {  
    public static void main(String[] args) {  
        Scanner scan = new Scanner(System.in);  
        System.out.println("Enter your name");  
        String name = scan.nextLine();  
  
        System.out.println("Enter your surname");  
        String surname = scan.nextLine();  
  
        System.out.println("Enter your email");  
        String email = scan.nextLine();  
  
        System.out.println("Enter your password");  
        String password = scan.nextLine();  
  
        scan.close();  
    }  
}
```



TECHPROED

DATABASE (VERİTABANI) NEDİR?



Veritabanı genellikle elektronik olarak bir bilgisayar sisteminde depolanan yapılandırılmış(**Structured**) bilgi veya veriden oluşan düzenli bir koleksiyondur.

Veritabanı genellikle bir Veritabanı Yönetim Sistemi DBMS (**DataBaseManagementSystem**) ile kontrol edilir.

Çoğu veritabanında veri yazma ve sorgulama için yapılandırılmış sorgu dili SQL (**Structured Query Language**) kullanılır.

DATABASE'IN FAYDALARI NELERDIR

- 1) Yuksek miktarda bilgi depolanabilir
- 2) Olusturma, Okuma,Degistirme ve Silme kolayligi
Create,Read,Update,Delete (CRUD)
- 3) Girisin kolay ve kontrollu olmasi
- 4) Dataya ulasim kolayligi
- 5) Guvenlik



no	adi	soyadi	dyeri	bid
1	Ali	Turan	İstanbul	1
2	Ahmet	Büyük	Ankara	1
3	Leyla	Şahin	İzmir	1
4	Can	Türkoğlu	Manisa	2
5	Aziz	Keskin	İstanbul	2
6	Talat	Şanlı	İzmir	3
7	Kamuran	Kece	Adana	3
8	Turgut	Cemal	Bursa	4



TECH

CODE

DATABASE VALIDATION(DOGRULAMA) TESTİ

Kaydol
Hızlı ve kolaydır.

Adın Soyadın

Cep telefonu numarası veya e-posta

Yeni şifre

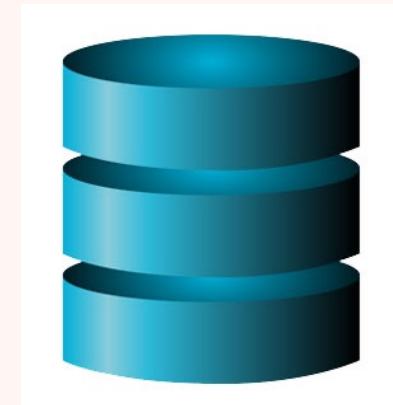
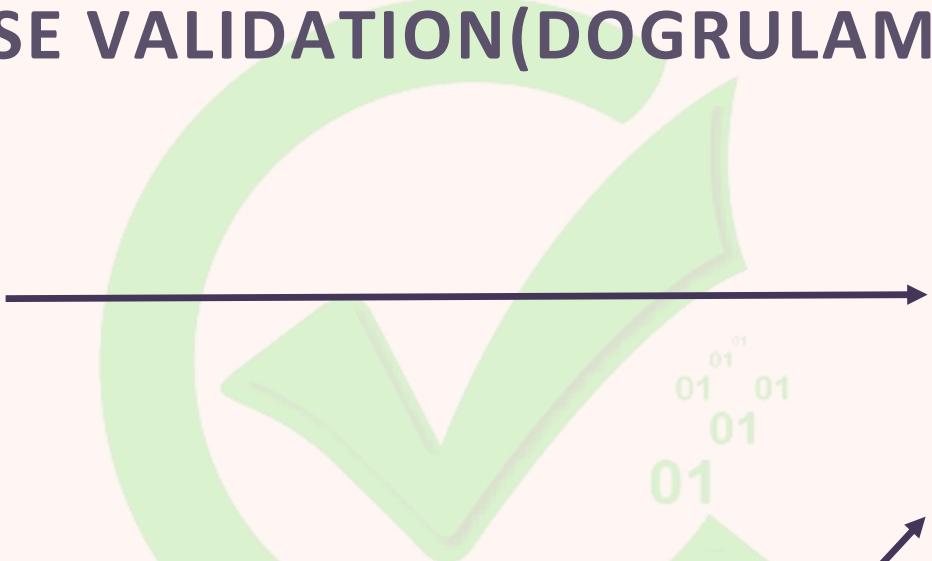
Doğum Tarihi Eki 2020

Cinsiyet Kadın Erkek Özel

Kaydol düğmesine tıklayarak, Koşullarımıza, Veri İkemizi ve Çerezler İkemizi kabul etmiş olursun. Bizden SMS Bildirimleri alabilir ve bu bildirimleri istediği zaman durdurabilirsin.

Kaydol

User Interface



Database

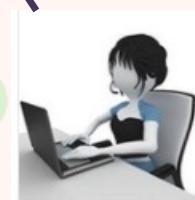
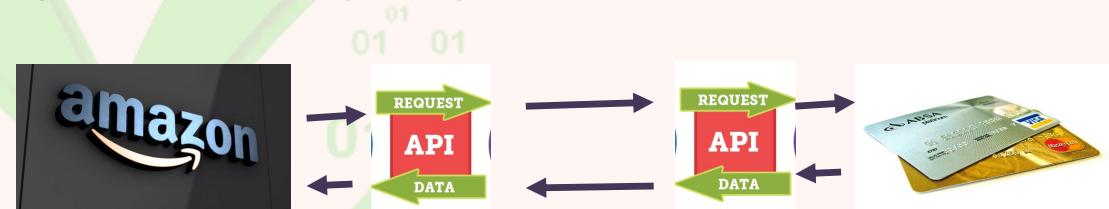
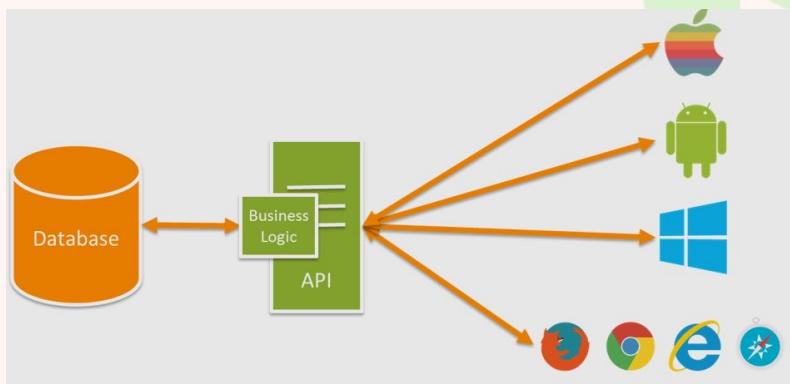


API

T E C H I R O E D

API

Application Programming Interface, bir uygulamaya ait yeteneklerin, başka bir uygulamada da kullanılabilmesi için, yeteneklerini paylaşan uygulamanın sağladığı arayüzdür.



TECH PRO E D

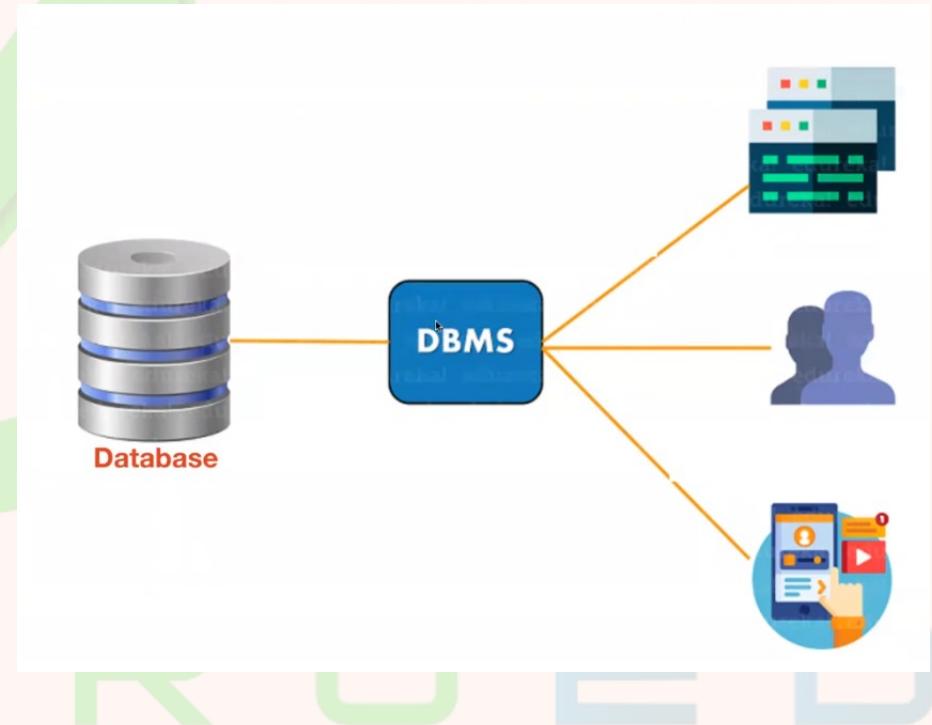
END To END (E2E) Testing

- 1 Eger datayi User Interface (UI) kullanarak yolladiysaniz
 - A) Datayi UI dan arama fonksiyonunu kullanarak doğrula (Selenium)
 - B) Datayi SQL kodlarini kullanarak doğrula (SQL + Selenium)
 - C) Datayi API kodlarini kullanarak doğrula (API + Selenium)
- 2) Eger datayi SQL kodlarini kullanarak yolladiysaniz
 - A) Datayi UI dan arama fonksiyonunu kullanarak doğrula (Selenium)
 - B) Datayi SQL kodlarini kullanarak doğrula (SQL + Selenium)
 - C) Datayi API kodlarini kullanarak doğrula (API + Selenium)
- 3) Eger datayi API kodlarini kullanarak yolladiysaniz
 - A) Datayi UI dan arama fonksiyonunu kullanarak doğrula (Selenium)
 - B) Datayi SQL kodlarini kullanarak doğrula (SQL + Selenium)
 - C) Datayi API kodlarini kullanarak doğrula (API + Selenium)

Data Base Management System (DBMS)

Veritabanlarını yönetmek, kullanmak, geliştirmek ve bakımını yapmak için kullanılan yazılımlara denir.

- Database'e erisimi düzenler
- Create,Read,Update,Delete işlemlerini düzenler
- Data güvenliğini sağlar
- Formlar oluşturur ve formları işler,
- Sorgular oluşturur ve sorgular iletilir,
- Raporlar oluşturur ve raporları işletir,
- Uygulamayı kontrol eder
- Diger uygulamalarla (Application) iletişimini saglar.



TABLOLAR (TABLES)

Headers====>
Row (Record) =====>
Row (Record) =====>
Row (Record) =====>
Row (Record) =====>

contactID	name	company	email
1	Bill Gates	Microsoft	bill@XBoxOneRocks.com
2	Steve Jobs	Apple	steve@rememberNewton.com
3	Linus Torvalds	Linux Foundation	linus@gnuWho.org
4	Andy Harris	Wiley Press	andy@aharrisBooks.net

Column (Field) =====>

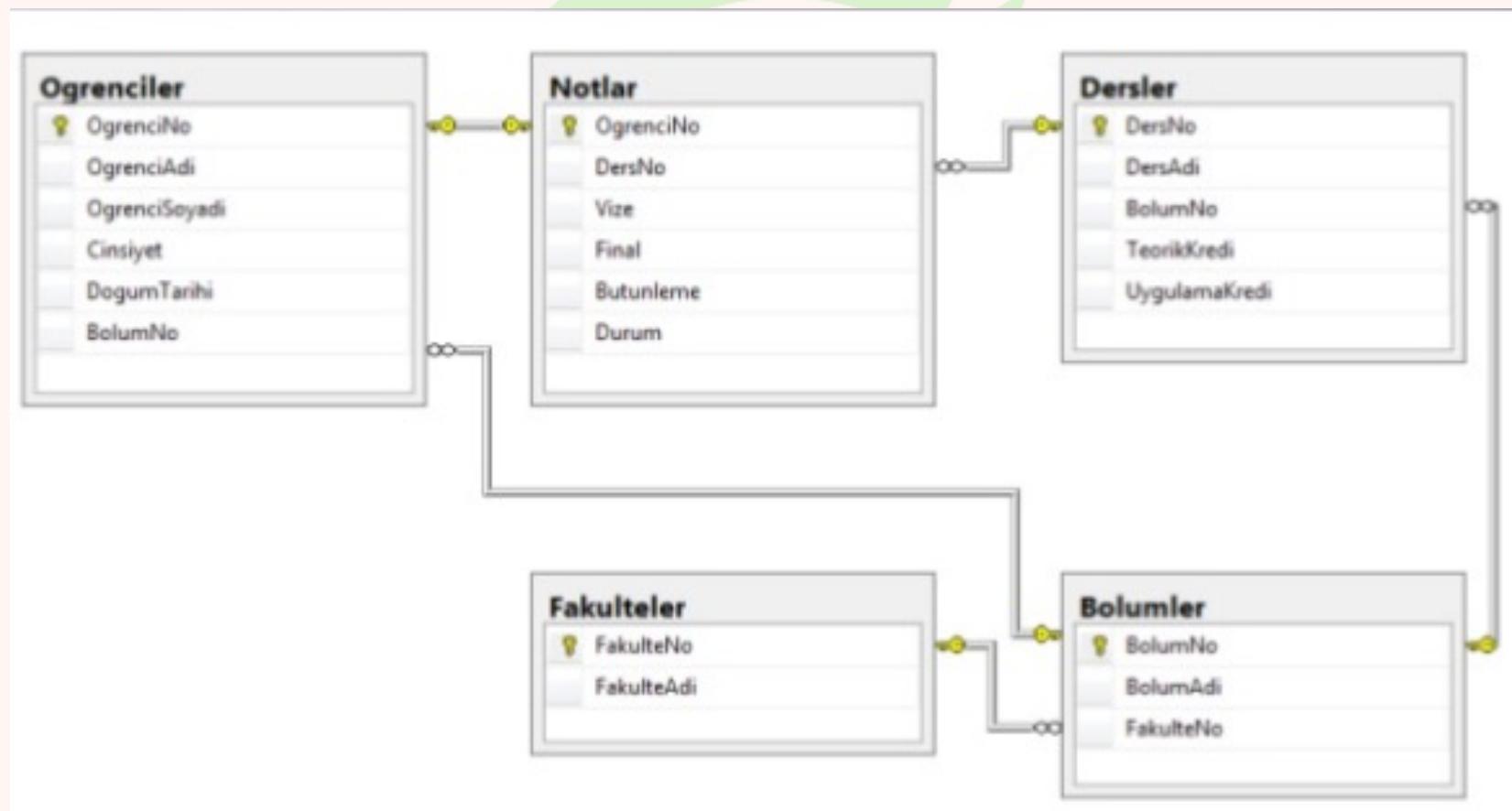
Column (Field) =====>

Column (Field) =====>

Column (Field) =====>

T E C H P R O E D

RELATIOANL DATABASES (ILISKILI TABLOLAR)



RELATIONAL DATABASES (ILISKILI TABLOLAR)

➤ **SQL tablolar** datalari ilişkili tablolarda depolar.

➤ Tablolar arasi ilişkiler net olmalıdır.

➤ Tablolar arası geçiş kolay olmalıdır

➤ Tabloların ve ilişkilerin bütünlüğüne
SCHEMA denir

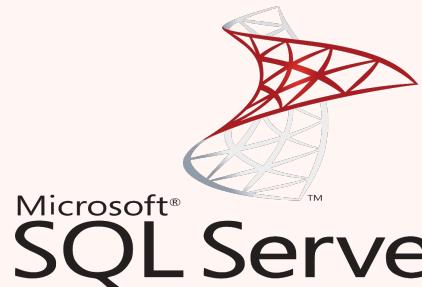
id	ogrenci_adi	ogrenci_soyadi
1	Elif	Türkmen
2	Ayşe	Sarı
3	Ender	Kaya
4	Ali	Demir
5	Adem	Salih

id	ogrenci_id	ders_id
1	1	3
2	1	5
3	2	1
4	3	4
5	4	2
6	4	3

id	ders_adı
1	Matematik
2	Tarih
3	Edebiyat
4	Yazılım
5	İstatistik

➤ **Relational Databases, SQL Databases (Structured Query Language) olarak**
da adlandırılır

Cok Kullanilan Relational Databases(SQL Database)

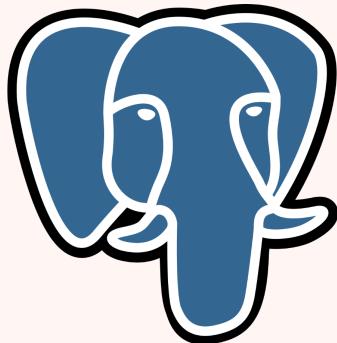


SQL Server : Microsoft tarafindan gelistirilmistir

Negatif: Pahali – Kurumsal Kullanicilar icin binlerce dolar odenmesi gereklidir
Pozitif : Zengin bir **user interface**'e sahip ve cok buyuk datalarin kullanilmasinda sorunsuz calisir.



Cok Kullanilan Relational Databases(SQL Database)



PostgreSQL Server : Created by a computer science professor Michael Stonebraker.

Negatif: Kurulum ve ayarlar zor. Yeni baslayanlar icin kullanimi zor
Pozitif: Yeni nesil olarak ortaya ciktig. Kisisellestirme mumkundur, zor gorevler icin ideal olabilir.

ORACLE
DATABASE

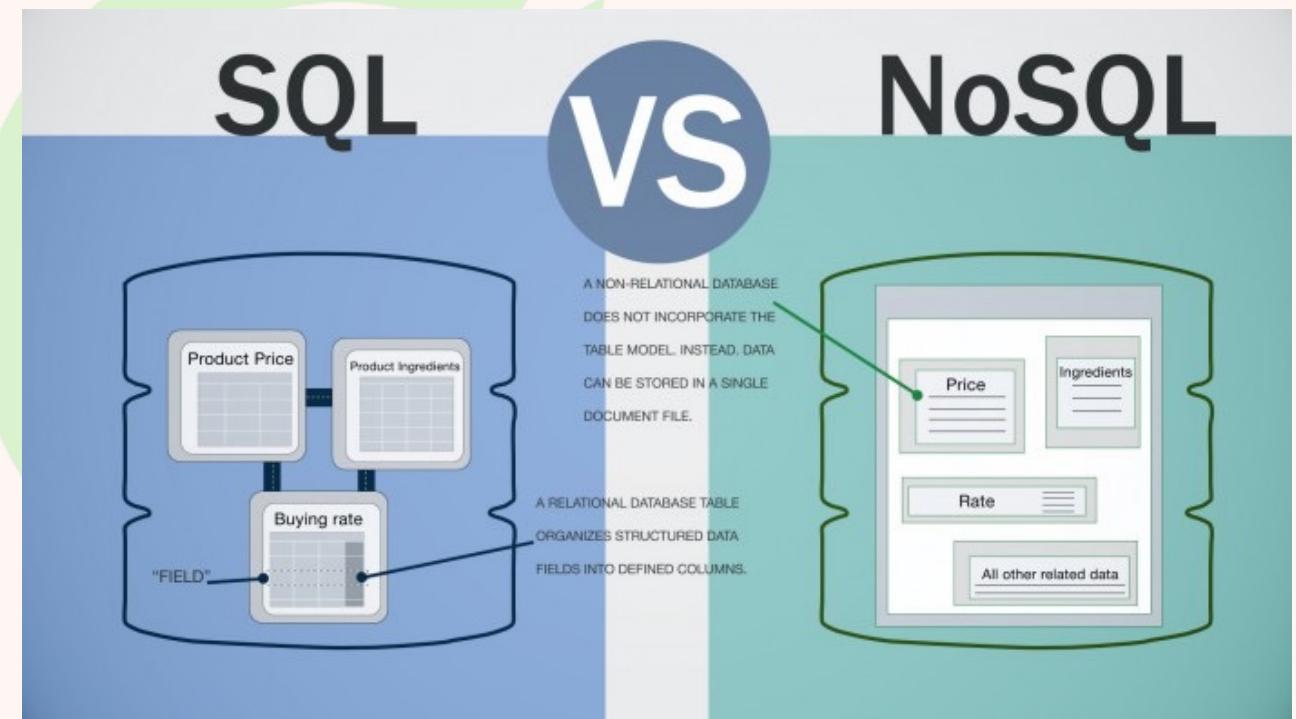


PL/SQL Oracle database sunuculari icinde depolanir
PL/SQL SQL komutlarini ozellikle karsilamak uzere dizayn edilmistir.

Pros: PL/SQL yüksek guvenlik seviyesi saglar ve Object-Oriented Programming'e uyumludur

Non Relational Databases(non-SQL Database)

SQL veritabanı verilerle ilgilenirken Yapısal Sorgu Dili kullanır. Veri yapısını belirlemek için önceden tanımlanmış şemalar gerektirir.



NoSQL veritabanı verilerle çalışırken Yapılandırılmamış Sorgu Dili kullanır.

SQL Komutları

SQL komutları 4 ana gruba ayrılır:

1. Veri Sorulama Dili (Data Query Language - DQL)

DQL içindeki SELECT komutu ile veritabanında yer alan **mevcut kayıtların** bir kısmını veya tamamını tanımlanan koşullara bağlı olarak alır.

SELECT : Veritabanındaki verileri alır.

2. Veri Kullanma Dili (Data Manipulation Language - DML)

DML komutları ile veritabanlarında bulunan verilere işlem yapılır. DML ile veritabanına yeni kayıt ekleme, mevcut kayıtları güncelleme ve silme işlemleri yapılır.

INSERT : Veritabanına yeni veri ekler.

UPDATE : Veritabanındaki verileri günceller.

DELETE : Veritabanındaki verileri siler.

SQL Komutları

3. Veri Tanımlama Dili (Data Definition Language - DDL)

DDL komutları ile veritabanı ve tabloları oluşturma, değiştirme ve silme işlemleri yapılır:

CREATE : Bir veritabanı veya veritabanı içinde tablo oluşturur.

ALTER : Bir veritabanı veya veritabanı içindeki tabloyu günceller.

DROP : Bir veritabanını veya veritabanı içindeki tabloyu siler.

4. Veri Kontrol Dili (Data Control Language - DCL)

DCL komutları ile kullanıcılaraya veritabanı ve tablolar için yetki verilir veya geri alınır:

GRANT : Bir kullanıcıya yetki vermek için kullanılır.

REVOKE : Bir kullanıcıya verilen yetkiyi geri almak için kullanılır.

Primary Key

Primary Keys



StudentId	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042
L0023487	Peter	Murray	P301
L0018453	Anne	Norris	S042

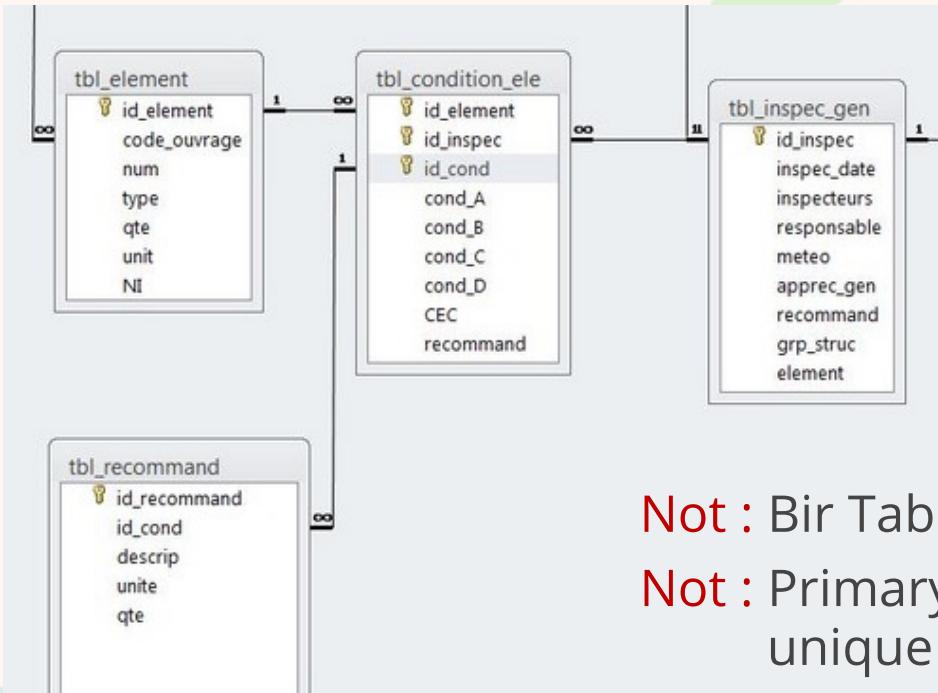
Primary Key (birincil anahtar), bir veri tablosunda yer alan her satır için bir vekil / tanımlayıcı (identify) görevi görür, kısıtlamadır (constraint) ve eşsizdir (Unique).

Satırlara ait değerlerin karışmaması adına bu alana ait bilginin tekrarlanmaması gereklidir.

Çoğunlukla tek bir alan (id, user_id, e_mail, username, national_identification_number vb.) olarak kullanılsa da birden fazla alanın birleşimiyle de oluşturulabilir

T E C H P R O E D

Primary Key



Primary Key değeri boş geçilemez ve NULL değer alamaz.

Relational veri tabanlarında (relational database management system) mutlaka birincil anahtar olmalıdır.

Not : Bir Tabloda en fazla 1 tane primary Key olabilir.

Not : Primary Key benzersiz (Unique) olmalıdır ama her unique data Primary Key degildir

Not : Primary key her türlü datayı içerebilir. Sayı, String..

Not : Her tabloda Primary Key olması zorunlu degildir

Primary Key

Primary Key, dış dünyadaki gerçek verileri temsil ediyorsa, örneğin; TC kimlik numarası, bir kitabın ISBN numarası, bir ürünün ismi, email hesabı gibi buna **Natural key** denir

StudentID	FirstName	LastName
10 ←	John	Walker
11	Tom	Hanks
12	Kevin	Star
13 ←	Carl	Wall
14	Andrei	Apazniak
15	Mark	High
16	Clara	Star
17	John	Ocean
18 ←	John	Walker
19	Pamela	Star
20 ←	Carl	Wall

Genel olarak kayıt eklenmeden önce üretilen sıra numarası gibi sayısal değerlere **Surrogate Key** denir

Email	FirstName	LastName
JWalker@gmail.com	John	Walker
THanks@gmail.com	Tom	Hanks
KStar@gmail.com	Kevin	Star
CWall@gmail.com	Carl	Wall
AApazniak@gmail.com	Andrei	Apazniak
MHigh@gmail.com	Mark	High
CStar@gmail.com	Clara	Star
JOcean@gmail.com	John	Ocean
JWalker01@gmail.com	John	Walker
PStar@gmail.com	Pamela	Star
CWall01@gmail.com	Carl	Wall

Foreign Key

Foreign Key iki tablo arasında relation olusturmak icin kullanılır

Foreign Key baska bir tablodaki Primary Key ile ilişkilendirilmiş olmalıdır

StudentID	FirstName	LastName	CourseID
10	John	Walker	200
11	Tom	Hanks	400
12	Kevin	Star	400
13	Carl	Wall	200
14	Andrei	Apazniak	300
15	Mark	High	400
16	Clara	Star	100
17	John	Ocean	100
18	John	Walker	200
19	Pamela	Star	300
20	Carl	Wall	NULL

Child Table

CourseID	CourseName	CourseCredit	CourseFee
100	Biology	3	1200
200	Math	3	1200
300	English	2	600
400	Selective	1	200

Parent Table

Bir Tabloda birden fazla Foreign Key olabilir

Foreign Key **NULL** degeri Kabul eder

Foreign Key olarak tanımlanan field'da **tekrarlar** olabilir

Foreign Key, değerleri farklı bir tablodaki Primary Key ile eşleşen bir sütun veya sütunların birleşimidir.

Foreign and Primary Key

Note: Foreign key Tablonun kendi icinde bir relation olusturabilir.

Emp_ID	first_name	last_name	birth_date	Gender	salary	Job_ID	Manager_ID
100	Jan	Levinson	1961-05-11	F	110,000	1	NULL
101	Michael	Scott	1964-03-15	M	75,000	2	100
102	Josh	Porter	1969-09-05	M	78,000	3	100
103	Angela	Martin	1971-06-25	F	63,000	2	101
104	Andy	Bernard	1973-07-22	M	65,000	3	101

Job_ID	Job_Name
2	SDET
3	Manual Tester
1	QE Lead

- 1) Michael Scott'un yoneticisi kimdir?
- 2) Angela Martin'in Job_Name'i nedir ?
- 3) Manual Tester'larin ortalama Salary'si ne kadardir ?
- 4) En yüksek Salary'yi alan kisinin Job_Name'i nedir?

SQL Composite Key

Job_ID	Job_Name
2	SDET
3	Manuel Tester
1	QA Led

Job Table

Recruiter	NumberOfClient
Mark Eye	121
John Ted	283
Angela Star	301
Cory AI	67

Recruiter Table

Job_Id	Name	Company
2	Mark Eye	RCG
3	John Ted	RCG
1	Mark Eye	Signature
1	John Ted	Info Log
1	Cory AI	Info Log
2	Angela Star	Signature

Company Table

Composite Key birden fazla field(kolon)'in kombinasyonu ile olusturulur.

Tek basina bir kolon Primary Key olma ozelliklerini tasimiyorsa, bu ozellikleri elde etmek icin birden fazla kolon birlestirilerek Primary olusturulur

“UNIQUE KEY” & “PRIMARY KEY”

“UNIQUE KEY” ve “PRIMARY KEY” arasındaki farklar

Primary Key

Bir Tabloda enfazla 1 tane olur
NULL deger Kabul etmez

Unique Key

Bir tabloda birden fazla olabilir
NULL degeri Kabul eder

“UNIQUE KEY” ve “PRIMARY KEY” ortak ozellikleri

Dublication(Cift Kullanim)'a izin vermez

T E C H P R O E D

Ornek Okul Tablosunun Bir Parcasi

sinif tablosu		
sinif id	sinif	sube adi
9a	9 a	
9b	9 b	
9c	9 c	
9d	9 d	
10a	10 a	
10b	10 b	
10c	10 c	

ders tablosu	
ders id	ders adi
k10	10.sinif kimya
k11	11.sinif kimya
k12	12.sinif kimya
b10	10.sinif biyoloji
k9	9.sinif kimya
b9	9.sinif biyoloji

ogrenci tablosu				
ogrenci no	adi	soyadi	giris yili	sinif id
111	ali	velioglu	2020	9a
112	ayse	atakul	2018	9a
113	hasan	delioglan	2019	9a
114	hulya	kar	2019	9b
115	ali	yasa	2019	9b
116	ayse	atakul	2020	9b
117	kemal	velioglu	2018	10a
118	hatice	gulsen	2019	10b
119	hasan	delioglan	2019	10c
120	kemal	kar	2018	10c

ogrenci sahisi bilgileri					
ogrenci no	tel	boyu	kilosu	saglik raporu	fotografi
111	12124435	160	50 var	var	

veli bilgileri						
ogrenci no	veli adi	veli soy	veli yak.	veli tel	veli tel 2	adres
111	hasan	velioglu	babasi	64654613	31646	

ogretmen tablosu			
adi	soyadi	ders	ogr id
ahmet	baba	kimya	k101
mehmet	kilim	fizik	f102
ayse	gulcu	tarih	t101
ayse	gulmez	biyoloji	b102
kemal	yasa	biyoloji	b105
fatma	yasa	kimya	k103

yazili tablosu				
ogrenci no	ders	ogretmen	not	
111	k9	k101	85	
112	b9	b102	80	
116	b9	b105	65	
118	k10	k103	90	

Related Tablolarla Calisma

One to One Relation

StudentID	FirstName	LastName
10	John	Walker
11	Tom	Hanks
12	Kevin	Star
13	Carl	Wall
14	Andrei	Apazniak
15	Mark	High
16	Clara	Star
17	John	Ocean
18	John	Walker
19	Pamela	Star
20	Carl	Wall

StudentID	Street	ZipCode	City	State
10	1234 W 23th Street	33018	Hialeah	Florida
11	1235 N 3th Street	22145	Austwell	Texas
12	1236 SE 12th Street	54234	Orange	California
13	1237 N 5th Street	33018	Hialeah	Florida
14	1238 SW 53th Street	33026	Miami	Florida
15	1239 S 123th Street	22314	Avery	Texas
16	1240 N 1 st Street	12345	Arlington	Virginia
17	1241 NW 2nd Street	65432	Pittsburgh	Penssylvania
18	1242 W 5th Street	22133	Baytown	Texas
19	1243 SE 55th Street	74352	Beachwood	Ohio
20	1244 SW 17th Street	22314	Avery	Texas

- 1) Tom Hanks'in adresi nedir?
- 2) John Walker'in eyaleti nedir?
- 3) ID'si 17 olan kisinin sehri nedir?

Related Tablolarla Calisma

One to Many Relation

CourseID	CourseName	CourseCredit	CourseFee	InstructorID
100	Biology	3	1200	1
200	Math	3	1200	2
300	English	2	600	3
400	Selective	1	200	1

StudentID	FirstName	LastName	CourseID
10	John	Walker	200
11	Tom	Hanks	400
12	Kevin	Star	400
13	Carl	Wall	200
14	Andrei	Apazniak	300
15	Mark	High	400
16	Clara	Star	100
17	John	Ocean	100
18	John	Walker	200
19	Pamela	Star	300
20	Carl	Wall	400

- 1) Biology dersi alan ogrenciler kimler?
- 2) Selective ders alan ogrencilerin isimleri ?
- 3) CourseFee 600 olan ogrencilerin isimleri ?

Related Tablolarla Calisma

Many to Many Relation

StudentID	FirstName	LastName
10	John	Walker
11	Tom	Hanks
12	Kevin	Star
13	Carl	Wall
14	Andrei	Apazniak
15	Mark	High
16	Clara	Star
17	John	Ocean
18	John	Walker
19	Pamela	Star
20	Carl	Wall

StudentID	InstructorID
12	1
11	2
12	2
13	1
15	1
17	3
15	4

InstructorID	FirstName	LastName	Phone	Department
1	Mark	Adam	1234567891	Science
2	Eve	Sky	1239876543	Engineering
3	Leo	Ocean	1237845691	Language
4	Andy	Mark	1232134567	Health

- 1) Ogretmeni Mark Adam olan ogrencilerin isimleri nedir?
- 2) Kevin Star'in ogretmenlerinin isimleri nedir?
- 3) Pamela Star'in ogretmenlerinin isimleri nedir?

SQL Data Types

String Data Types

Data Type	Açıklama
char(size)	Maximum boyutu 2000 byte olur. 1 karakter 1 byte kullanır. “size” database’e eklenecek karakter sayisidir. “char” data tipinden uzunluğu sabit dataları depolar. (Strings) “char” SSN, zip kodu gibi uzunluğu sabit dataları depolamak için idealdir.
nchar(size)	Maximum boyutu 2000 byte olur. 1 karakter 2 byte kullanır “size” depolanacak karakter sayısıdır. “nchar” Unicode dataları depolamak için kullanılır. Genellikle farklı dillerdeki karakterler için kullanılır Uzunluğu belli Stringler için kullanılır.
varchar2(size)	Maximum boyutu 4000 byte olur. 1 karakter 1 byte kullanır. “size” database’e eklenecek max. karakter sayisidir. Degisken uzunluktaki stringler için kullanılır.
nvarchar2(size)	Maximum boyutu 8000 byte olur. 1 karakter 2 byte kullanır “size” depolanacak karakter sayısıdır. Degisken uzunluktaki stringlerin Unicode değerleri için kullanılır.

Alphabets	CHAR (4)	Data Size	VARCHAR (4)	Data Size
''	' '	4 bytes	''	1 byte
'ab'	'ab '	4 bytes	'ab'	3 bytes
'abcd'	'abcd'	4 bytes	'abcd'	5 bytes
'abcdefg'	'abcd'	4 bytes	'abcd'	5 bytes

SQL Data Types

Numeric Data Types

Data Type	Açıklama
number(p, s)	<p>“Precision” (p) sayidaki rakam sayisidir “Scale” (s) virgulden sonar kac rakam oldugunu belirler Ornegin: 1234,56 ==> Precision : 6, Scale : 2.</p> <p>Precision (p) can range from 1 to 38 Scale (s) can range from -84 to 127</p> <p>1) “number(5, 2)” virgulden once 3,vigulden sonra 2 rakam olan sayi ==> 123,45</p> <p>2) “number(4, 2)” ==> 123,45 ==> error verir</p> <p>3) “number(7)” ondalik kismi olmayan 7 basamakli sayı demektir ==> 12345,67’i kabul eder ama 12345 olarak depolar Note: “number(7)” ve “number(7, 0)” ayni seydir</p> <p>4) “number(7, -2)” rounds the numeric value to hundreds. ==> 1234567,89 ==> 1234600</p>

SQL Data Types

Numeric Data Types

DBMS Numeric Types:

<i>DBMS and version</i>	<i>Types</i>
MySQL 5.7	INTEGER(TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT, INTEGER) FIXED-POINT(DECIMAL, NUMERIC) FLOATING-POINT(FLOAT, DOUBLE) BIT-VALUE(BIT),
PostgreSQL 9.5.3	SMALLINT, INTEGER, BIGINT, DECIMAL, NUMERIC, REAL, DOUBLE PRECISION, SMALLSERIAL, SERIAL, BIGSERIAL
SQL Server 2014	EXACT NUMERICS(BIGINT, BIT, DECIMAL, INT, MONEY, NUMERIC, SMALLINT, SMALLMONEY, TINYINT) APPROXIMATE NUMERICS(FLOAT, REAL)
Oracle 11g	NUMBER FLOATING-POINT(BINARY_FLOAT, BINARY_DOUBLE)

TECH PRO E D

SQL Data Types

Date Data Types

<i>Data Type</i>	<i>Acıklama</i>
DATE	<p>“DATE” data tipi tarih ve zamani depolamak için kullanılır. Saniyenin virgullu kısmını da alır. “DATE” yıl, ay, gün, saat, dakika, ve saniye içerir.</p> <p>Standart “Date Format”, “dd - MMM - yy”. Örneğin 13 - Apr - 20</p> <p>Tarih formatını “ALTER SESSION SET NLS_DATE_FORMAT = “YYYY-MM-DD” kodu kullanılarak değiştirebilir. Koddan sonra tarih 2020 - 04 - 13 olur.</p>

T E C H P R O E D

SQL Data Types

BLOB Data Types

<i>Data Type</i>	<i>Acıklama</i>
BLOB	“BLOB” , “Binary Large OBjects” demektir “BLOB” resim,video,ses gibi dataları binary formatına çevirerek depolar.

T E C H P R O E D

SQL Komutları

4. Veri Kontrol Dili (Data Control Language - DCL)
veritabanı ve tablolar için yetki verilir
veya geri alınır

GRANT : Bir kullanıcıya yetki
vermek için kullanılır.

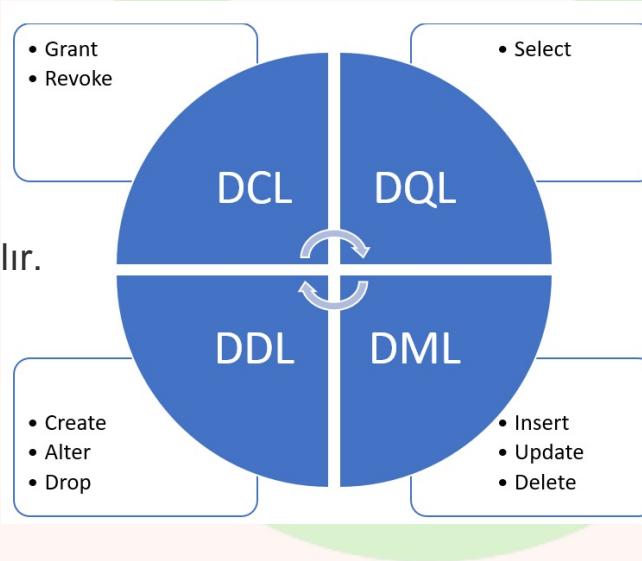
REVOKE : Bir kullanıcıya verilen
yetkiyi geri almak için kullanılır.

3. Veri Tanımlama Dili
(Data Definition Language - DDL)
veritabanı ve tabloları oluşturma,
değiştirme ve silme işlemleri yapılır

CREATE : Bir veritabanı veya tablo oluşturur.

ALTER : Bir veritabanı veya tabloyu günceller.

DROP : Bir veritabanını veya tabloyu siler.



1. Veri Sorgulama Dili (Data Query Language - DQL)
mevcut kayıtların bir kısmını veya tamamını
tanımlanan koşullara bağlı olarak alır.

SELECT : Veritabanındaki verileri alır.

2. Veri Degistirme Dili (Data Manipulation
Language - DML)
veritabanına yeni kayıt ekleme, mevcut
kayıtları güncelleme ve silme işlemleri
yapılır.

INSERT : Veritabanına yeni veri ekler.

UPDATE : Veritabanındaki verileri günceller.

DELETE : Veritabanındaki verileri siler.

Table Nasil Olusturulur?

1) Create from Scratch

```
CREATE TABLE ogrenciler  
(  
id char(4),  
isim_soyisim varchar2(50),  
not_ort number(5,2),  
kayit_tarihi date  
);
```

2) Var olan tablodan yeni tablo olusturmak

```
CREATE TABLE ogrenci_ortalamalar  
AS SELECT isim_soyisim, not_ort  
FROM ogrenciler;
```

T E C H P R O E D

Tabloya Nasil Data Eklenir (INSERT INTO)?

- **INSERT INTO** komutu, Oracle SQL'de tabloya bir veya birden fazla kayıt eklemek için kullanılır.

Olusturdugumuz ogrenciler tablosuna kayıt ekleyelim

1) Tum Field'lere data eklemek icin

INSERT INTO ogrenciler **VALUES** (1234, 'Ali Can', 85.60, '14-Oct-2020');

2) Bazi Field'lere data eklemek icin

INSERT INTO ogrenciler(ogrenci_id,isim_soyisim) **VALUES** (123456789, 'Ali Can');

T E C H P R O E D

Table Nasıl Olusturulur?

1) Create from Scratch

Practice1:

“tedarikciler” isminde bir tablo olusturun ve “tedarikci_id”, “tedarikci_ismi”, “tedarikci_adres” ve “ulasim_tarihi” field’lari olsun. Tablo olusturduktan sonra 3 tane kayit ekleyelim.

```
CREATE TABLE tedarikciler  
(  
    tedarikci_id char(10),  
    tedarikci_ismi varchar2(50),  
    tedarikci_adres varchar2(100),  
    ulasim_tarihi date  
)
```

2) Var olan tablodan yeni tablo olusturmak

“tedarikciler_id_name” isminde bir tabloyu “tedarikciler” tablosundan olusturun. Icinde “tedarikci_id”, “tedarikci_ismi” field’lari olsun.

```
CREATE TABLE tedarikci_ziyaret  
AS  
SELECT tedarikci_ismi,ulasim_tarihi  
FROM tedarikciler;
```

Tabloya Nasil Data Eklenir (INSERT INTO)?

Practice 1:

“*ogretmenler*” isminde bir SQL tablosu olusturun. Icinde “kimlik_no”, “isim”, “brans” ve “cinsiyet” field’lari olsun.

“*ogretmenler*” tablosuna bilgileri asagidaki gibi olan bir kisi ekleyin.

Kimlik_no: 234431223, *isim*: Ayse Guler, *brans* : Matematik, *cinsiyet*: kadin.

`CREATE TABLE ogretmenler`

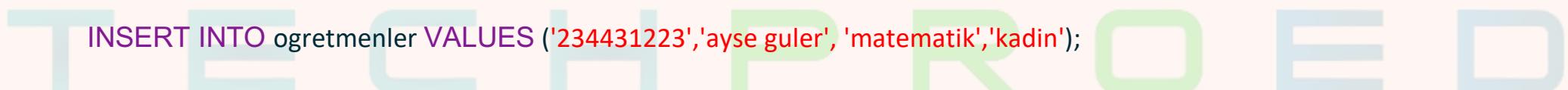
```
(  
kimlik_no char(11),  
isim varchar2(50),  
brans varchar2(50),  
cinsiyet varchar2(50)  
);
```

Ekleme :

“*ogretmenler*” tablosuna bilgileri asagidaki gibi olan bir kisi ekleyin.

Kimlik_no: 567597624, *isim*: Kemal Yasa

```
INSERT INTO ogretmenler VALUES ('234431223','ayse guler', 'matematik','kadin');
```



Bir field'in “tekrarsiz” deger almasi nasil saglanir?

“id” field'ini “tekrarsiz” yapmak icin , id field'inda Data Type'dan sonra “UNIQUE” yazmak gereklidir

```
CREATE TABLE ogrenciler2  
(  
    id char(4) UNIQUE,  
    isim varchar2(50),  
    not_ortalamasi number(5,2),  
    adres varchar2(100),  
    kayit_tarihi date  
)
```



T E C H P R O E D

Bir field'in “NULL” deger almaması nasıl saglanır?

“isim” field’inin “NULL” deger kabul etmemesi icin , isim field’i olusturulurken Data Type’dan sonra “NOT NULL” yazmak gereklidir

```
CREATE TABLE ogrenciler3  
(  
    id char(4) UNIQUE,  
    isim varchar2(50) NOT NULL,  
    not_ortalamasi number(5,2),  
    adres varchar2(100),  
    kayit_tarihi date  
)
```

T E C H P R O E D

Tabloya Nasil Data Eklendir (INSERT INTO)?

Note: INSERT INTO kodunu kullanarak bir tabloya data eklemek istediginizde, CONSTRAINT'lere uymak zorundayiz. Ornegin; NOT NULL yazan kisma bir deger atamak zorundayiz

Ornek : Personel isminde bir tablo olusturun, icinde id,isim,soyisim,email,ise_baslama_tarihi ve maas fieldlari olsun, isim field'i bos bırakılamasin

```
CREATE TABLE personel  
(  
    id char(10),  
    isim varchar2(50) NOT NULL,  
    soyisim varchar2(50),  
    email varchar2(50),  
    ise_baslama_tar date,  
    maas number(6)  
) ;
```

INSERT INTO personel (id,is_unvani) VALUES(123456789, 'isci');

ORA-01400: cannot insert NULL into ("SQL_LFGUHVRSSOWWDACLEMHRMQGCJQ"."STUDENTS"."NAME")
ORA-06512: at "SYS.DBMS_SQL", Line 1721

Bir Tabloya “Primary Key” Nasıl Eklenir

- 1) Primary Key bir record'u tanımlayan bir field veya birden fazla field'in kombinasyonudur.
- 2) Primary Key Unique'dir
- 3) Bir tabloda en fazla bir Primary Key Olabilir
- 4) Primary Key field'ında hiç bir data NULL olamaz

“**id**” field'ini “primary key” yapmak için 2 yol var

- 1) Data Type'dan sonra “**PRIMARY KEY**” yazarak.

```
CREATE TABLE ogrenciler4
(
    id char(4) PRIMARY KEY,
    isim varchar2(50) NOT NULL,
    not_ortalamasi number(5,2),
    adres varchar2(100),
    kayit_tarihi date
);
```

Bir Tabloya “Primary Key” Nasıl Eklenir

2) CONSTRAINT Keyword Kullanılarak Primary Key Tanımlanabilir

“CONSTRAINT constraintName PRIMARY KEY(column1, column2, ... column_n)”

```
CREATE TABLE ogrenciler5  
(  
    id char(4),  
    isim varchar2(50) NOT NULL,  
    not_ortalamasi number(5,2),  
    adres varchar2(100),  
    kayit_tarihi date,  
    CONSTRAINT ogrenciler_pk PRIMARY KEY(id)  
);
```

Bir Tabloya “Primary Key” Nasıl Eklenir

Practice 2:

“sehirler” isimli bir Table olusturun. Tabloda “alan_kodu”, “isim”, “nufus” field’lari olsun. Isim field’i bos bırakılamasın.

1.Yontemi kullanarak “alan_kodu” field’ini “Primary Key” yapın

`CREATE TABLE sehirler`

```
(  
    alan_kodu number(3) PRIMARY KEY,  
    isim varchar2(20) NOT NULL,  
    nufus number(8)  
)
```

Practice 3:

“ogretmenler” isimli bir Table olusturun. Tabloda “id”, “isim”, “brans”, “cinsiyet” field’lari olsun. Id field’i tekrarli deger Kabul etmesin.
2.Yontemi kullanarak “id ve isim” field’lerinin birlesimini “primary key” yapın

`CREATE TABLE ogretmenler`

```
(  
    id char(10) UNIQUE,  
    isim varchar2(20),  
    brans varchar2(20),  
    cinsiyet varchar2(20)  
    CONSTRAINT ogretmenler_pk PRIMARY KEY (id,isim)  
)
```

Tabloya “Foreign Key” Nasil Eklenir ?

- Foreign Key iki tablo arasında Relation olusturmak icin kullanilir.
- Foreign Key baska bir tablonun Primary Key'ine baglidir.
- Referenced table (baglanilan tablo, Primary Key'in oldugu Tablo) *parent table* olarak adlandirilir. Foreign Key'in oldugu tablo ise *child table* olarak adlandirilir.
- Bir Tabloda birden fazla Foreign Key olabilir
- Foreign Key NULL degeri alabilir

Note 1: “Parent Table” olmayan bir id'ye sahip datayi “Child Table”a ekleyemezsınız

Note 2: Child Table'i silmeden Parent Table'i silemezsiniz. Once “Child Table” silinir, sonra “Parent Table” silinir.

T E C H P R O E D

Tabloya “Foreign Key” Nasıl Eklenir ?

SIRKETLER Tablosu

Primary Key

SIRKET_ID	SIRKET	PERSONEL SAYISI
100	Honda	12000
101	Ford	18000
102	Hyundai	10000
103	Toyota	21000

PERSONEL Tablosu

Foreign Key

ID	ISIM	SEHIR	MAAS	SIRKET
123456789	Ali Seker	Istanbul	2500	Honda
234567890	Ayse Gul	Istanbul	1500	Toyota
345678901	Veli Yilmaz	Ankara	3000	Honda
456789012	Veli Yilmaz	Izmir	1000	Ford
567890123	Veli Yilmaz	Ankara	7000	Hyundai
456789012	Ayse Gul	Ankara	1500	Ford
123456710	Fatma Yasa	Bursa	2500	Honda

Parent Table

Child Table

T E C H P R O E D

Tabloya “Foreign Key” Nasıl Eklenir ?

Practice 4:

“tedarikciler3” isimli bir tablo olusturun. Tabloda “tedarikci_id”, “tedarikci_ismi”, “iletisim_isim” field’lari olsun ve “tedarikci_id” yi Primary Key yapin.
“urunler” isminde baska bir tablo olusturun “tedarikci_id” ve “urun_id” field’lari olsun ve “tedarikci_id” yi Foreign Key yapin.

```
CREATE TABLE urunler
(
    tedarikci_id char(10),
    product_id char(10),
    CONSTRAINT urunler_fk FOREIGN KEY (tedarikci_id) REFERENCES tedarikciler (tedarikci_id) );
```

```
CREATE TABLE tedarikciler
(
    tedarikci_id char(10),
    tedarikci_ismi varchar2(50),
    iletisim_isim varchar2(50),
    CONSTRAINT tedarikci_pk PRIMARY KEY (tedarikci_id)
);
```

Tabloya “Foreign Key” Nasil Eklenir ?

Practice 5:

“*tedarikciler*” isimli bir Tablo olusturun. Icinde “*tedarikci_id*”, “*tedarikci_isim*”, “*iletisim_isim*” field’lari olsun. “*tedarikci_id*” ve “*tedarikci_isim*” fieldlarini birlestirerek Primary Key olusturun. “*urunler*” isminde baska bir tablo olusturun. Icinde “*tedarikci_id*” ve “*urun_id*” fieldlari olsun. “*tedarikci_id*” ve “*urun_id*” fieldlarini birlestirerek Foreign Key olusturun

```
CREATE TABLE tedarikciler  
(  
    tedarikci_id number(10),  
    tedarikci_isim varchar2(50),  
    iletisim_isim varchar2(50),  
    CONSTRAINT tedarikci_pk PRIMARY KEY  
        (tedarikci_id,tedarikci_isim)  
);
```

```
CREATE TABLE urunler  
(  
    tedarikci_id number(10),  
    urun_id varchar2(10),  
    CONSTRAINT fk_tedarikci FOREIGN KEY  
        (tedarikci_id,urun_id) REFERENCES  
        tedarikciler(tedarikci_id,tedarikci_isim));
```

Tabloya “CHECK Constraint” Nasıl Eklenir ?

CHECK ile bir alana girilebilecek değerleri sınırlayabiliriz.

Mesela tablomuzda YAŞ bir alanı number data tipinde yani sayısal alan olarak belirlemiş olabiliriz. Ancak bu alan negatif sayı girilmesi anlamsız olacağı için CHECK yapısını kullanarak negatif giriş yapılmasını engelleyebiliriz.

Ornek : sehirler2 tablosu olusturalim, nufusun negatif deger girilmemesi icin sinirlandirma (Constraint) koyalim

```
CREATE TABLE sehirler2  
(  
alan_kodu number(3) PRIMARY KEY,  
isim varchar2(20) NOT NULL,  
nufus number(8) CHECK (nufus>0)  
);
```

SELECT KOMUTU

1) Tablodaki Tum Field'lari Cagirma

```
CREATE TABLE ogrenciler  
(  
    id number(9),  
    isim varchar2(50),  
    adres varchar2(100),  
    yazili_not number(3)  
);
```

```
INSERT INTO ogrenciler VALUES(123, 'Ali Can', 'Ankara', 75);  
INSERT INTO ogrenciler VALUES(124, 'Merve Gul', 'Ankara', 85);  
INSERT INTO ogrenciler VALUES(125, 'Kemal Yasa', 'Istanbul', 85);
```

```
SELECT *  
FROM students;
```

Tablodaki tum dataları getirir

```
SELECT *  
FROM students  
WHERE yazili_not >80;
```

Tablodaki yazili_notu 80'den buyuk olan kayitlari getirir

T E C H P R O E D

WHERE ile Kullanılan Mantıksal Operatorler

- = ==> Equal to sign
- > ==> Greater than sign
- < ==> Less than sign
- >= ==> Greater than or equal to sign
- <= ==> Less than or equal to sign
- <> ==> Not Equal to sign
- AND** ==> And operator
- OR** ==> Or operator



T E C H P R O E D

SELECT KOMUTU

2) Tablodaki Belli Bir Field'i Çağırma

`SELECT adres
FROM students;`

Tablodan sadece adres field'indaki tüm
dataları getirir

ADRES
Ankara
Ankara
Istanbul

`SELECT adres
FROM students
WHERE yazili_not=85;`

Tablodan sadece yazılı notu 85
olanların adres field'indakidataları getirir

ADRES
Ankara
Istanbul

T E C H P R O E D

SELECT KOMUTU

3) Tablodan Birden Fazla Field'i Cagirma

**SELECT adres,isim
FROM ogrenciler;**

Tablodan adres ve isim field'indaki tum
datalari getirir

**SELECT adres,isim
FROM ogrenciler**

WHERE yazılı_not>80; Tablodan yazılı notu 80'den buyuk
olan kayitların adres ve isim
field'indaki dataları getirir

ADRES	ISIM
Ankara	Ali Can
Ankara	Merve Gul
Istanbul	Kemal Yasa

ADRES	ISIM
Ankara	Merve Gul
Istanbul	Kemal Yasa

T E C M I T R O L U K

SELECT KOMUTU

- 4) Tablodan Bir Kayida Ait Birden Fazla Field'i Cagirma

```
SELECT adres,isim  
FROM ogrenciler  
WHERE id=123 ;
```

Tablodan id'si 123 olan kaydin adres ve isim field'indaki
datalari getirir

T E C H P R O E D

Tablodaki Data Nasil Update Edilir (UPDATE SET)?

Mart_satislar isimli bir tablo olusturun, icinde urun_id, musteri_isim, urun_isim ve urun_fiyat field'lari olsun

```
CREATE TABLE mart_satislar  
(  
    urun_id number(10),  
    musteri_isim varchar2(50),  
    urun_isim varchar2(50),  
    urun_fiyat number(10)  
);
```

```
INSERT INTO mart_satislar VALUES (10, 'Ali', 'Honda',75000);  
INSERT INTO mart_satislar VALUES (10, 'Ayse', 'Honda',95200);  
INSERT INTO mart_satislar VALUES (20, 'Hasan', 'Toyota',107500);  
INSERT INTO mart_satislar VALUES (30, 'Mehmet', 'Ford', 112500);  
INSERT INTO mart_satislar VALUES (20, 'Ali', 'Toyota',88000);  
INSERT INTO mart_satislar VALUES (10, 'Hasan', 'Honda',150000);  
INSERT INTO mart_satislar VALUES (40, 'Ayse', 'Hyundai',140000);  
INSERT INTO mart_satislar VALUES (20, 'Hatice', 'Toyota',60000);
```

- 1) Ismi hatice olan musterinin urun_id'sini 30,urun_isim'ini Ford yapin
- 2) Toyota marka araclara %10 indirim yapin
- 3) Ismi Ali olanların urunfiyatlarını %15 artirin
- 4) Honda aracların urun kodu'nu 50 yapin

Tablodaki Data Nasil Update Edilir (UPDATE SET)?

1) Bir tedarikciler tablosu olusturun icinde id, isim ve iletisim_isim field'lari olsun. Id ve isim'i beraber Primary Key yapin

```
CREATE TABLE tedarikciler
(
    id number(10),
    isim varchar2(50),
    iletisim_isim varchar2(50),
    CONSTRAINT tedarikci_pk PRIMARY KEY (id, isim)
);
```

3) id'si 1 olan tedarikcinin ismini 'KRM' ve iletisim_isim'ini 'Kemal Kan' yapin

```
UPDATE tedarikciler
SET isim = 'KRM', iletisim_isim = 'Hasan Kan'
WHERE id =1;
```

2) Icine 3 kayit ekleyin (1, 'ACB', 'Ali Can'), (2, 'RDB', 'Veli Gul'), (3, 'KMN', 'Ayse Gulmez').

```
INSERT INTO tedarikciler VALUES (1, 'ACB', 'Ali Can');
INSERT INTO tedarikciler VALUES (2, 'RDB', 'Veli Gul');
INSERT INTO tedarikciler VALUES (3, 'KMN', 'Ayse Gulmez');
```

4) Ismi RDB olan tedarikcinin iletisim isim'ini Kemal Yasa yapin

```
UPDATE tedarikciler
SET iletisim_isim='Kemal Yasa'
WHERE isim='RDB';
```

TECHPROED

Tablodaki Data Nasil Update Edilir (UPDATE SET)?

Practice 11: verilen tablolara gore asagidaki islemleri yapin.

- a) Urnler tablosundan Ali Can'in aldigı ürünün ismini, tedarikci tablosunda irtibat_isim Merve Temiz olan şirketin ismi ile degistirin
- b) TV satın alan musterinin ismini, Apple'in irtibat_isim'i ile degistirin

```
CREATE TABLE tedarikci
(
    id number(5) PRIMARY KEY,
    isim varchar2(50),
    irtibat_isim varchar2(50)
);
```

```
CREATE TABLE urunler
(
    tedarikci_id number(5),
    urun_id number(11),
    urun_isim varchar2(50),
    musteri_isim varchar2(50),
    CONSTRAINT urunler_fk FOREIGN KEY(tedarikci_id) REFERENCES tedarikci(id )
);
```

```
INSERT INTO tedarikci VALUES(100, 'IBM', 'Ali Can');
INSERT INTO tedarikci VALUES(101, 'APPLE', 'Merve Temiz');
INSERT INTO tedarikci VALUES(102, 'SAMSUNG', 'Kemal Can');
INSERT INTO tedarikci VALUES(103, 'LG', 'Ali Can');
```

```
INSERT INTO urunler VALUES(100, 1001,'Laptop', 'Suleyman');
INSERT INTO urunler VALUES(101, 1002,'iPad', 'Fatma');
INSERT INTO urunler VALUES(102, 1003,'TV', 'Ramazan');
INSERT INTO urunler VALUES(103, 1004,'Phone', 'Ali Can');
```

Tablodaki Data Nasil Update Edilir (UPDATE SET)?

ID	ISIM	IRTIBAT_ISIM
100	IBM	Ali Can
101	APPLE	Merve Temiz
102	SAMSUNG	Kemal Can
103	LG	Ali Can

tedarikci

a) Urunler tablosundan Ali Can'in aldigı ürünün ismini, tedarikci tablosunda irtibat_isim Merve Temiz olan şirketin ismi ile degistirin

UPDATE urunler

```
SET urun_isim= (SELECT isim  
    FROM tedarikci  
   WHERE irtibat_isim='Merve Temiz')  
WHERE musteri_isim='Ali Can';
```

TEDARIKCI_ID	URUN_ID	URUN_ISIM	MUSTERIR_ISIM
103	1004	Phone	Ali Can
100	1001	Laptop	Suleyman
101	1002	iPad	Fatma
102	1003	TV	Ramazan

urunler

b) TV satin alan musterinin ismini, Apple'in irtibat_isim'i ile degistirin

UPDATE urunler
SET musteri_isim=(SELECT irtibat_isim
 FROM tedarikci
 WHERE isim='APPLE')
WHERE urun_isim='TV';

T E C H N O L O G I E S



SQL

Structured Query Language
Yapılandırılmış Sorgu Dili

15 Ocak 2021

DERS 05

Update Data

Delete Data

T E C H P R O E D

Mehmet Bulutluoz
Elektronik muh.

Tablodaki Data Nasil Update Edilir (UPDATE SET)?

- 1) Ogrenciler6 tablosu olusturun. Icinde id,isim,veli_isim ve not_ort field'lari olsun. Id field'i birlikte Primary Key olsun.

```
CREATE TABLE ogrenciler
(
    id char(3),
    isim varchar2(50),
    veli_isim varchar2(50),
    not_ort number(3),
    CONSTRAINT ogrenciler_pk PRIMARY KEY (id)
);
```

- 3) notlar tablosu olusturun. ogrenci_id,ders_adi,yazili_notu field'lari olsun, ogrenci_id field'i Foreign Key olsun

```
CREATE TABLE notlar
(
    ogrenci_id char(3),
    ders_adi varchar2(30),
    yazili_notu number (3),
    CONSTRAINT notlar_fk FOREIGN KEY (ogrenci_id) REFERENCES ogrenciler (id)
);
```

- 2) 3 kisiyi tabloya ekleyin. (123, 'Ali Can', 'Hasan',75), (124, 'Merve Gul', 'Ayse',85), (125, 'Kemal Yasa', 'Hasan',85).

```
INSERT INTO ogrenciler6 VALUES(123, 'Ali Can', 'Hasan',75);
INSERT INTO ogrenciler6 VALUES(124, 'Merve Gul', 'Ayse',85);
INSERT INTO ogrenciler6 VALUES(125, 'Kemal Yasa', 'Hasan',85);
INSERT INTO ogrenciler6 VALUES(126, 'Fatma Gun', 'Mehmet',90);
INSERT INTO ogrenciler6 VALUES(127, 'Ali Can', 'Mehmet',91);
```

- 4) notlar tablosuna 3 kayit ekleyin ('123','kimya',75), ('124','fizik',65),('125','tarih',90)

```
INSERT INTO notlar VALUES ('123','kimya',75);
INSERT INTO notlar VALUES ('124','fizik',65);
INSERT INTO notlar VALUES ('125','tarih',90);
INSERT INTO notlar VALUES ('126','Matematik',90);
INSERT INTO notlar VALUES ('127','Ingilizce',90);
```

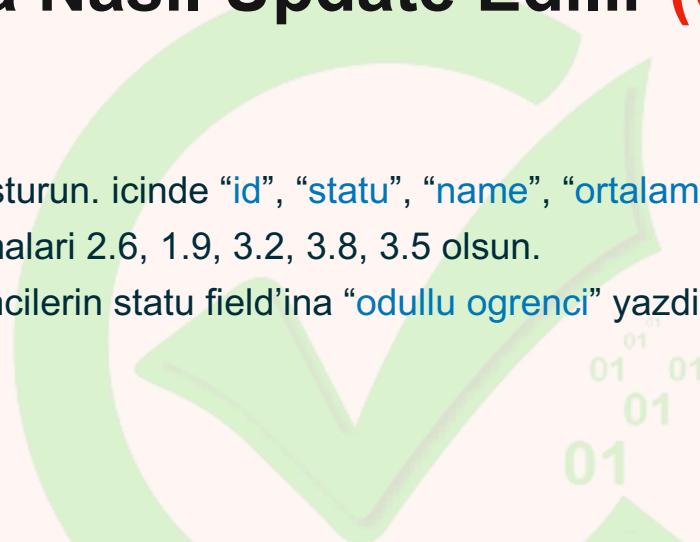
- 5) Tum ogrencilerin yazili notlarini notlar tablosundaki ile update edin

```
UPDATE ogrenciler
SET yazili_notu= (SELECT yazili_notu
                    FROM notlar
                   WHERE ogrenciler.id=notlar.ogrenci_id
)
WHERE id>100;
```

Tablodaki Data Nasil Update Edilir (UPDATE SET)?

Practice 9:

- a) "ogrenciler" isminde bir tablo olusturun. icinde "id", "statu", "name", "ortalama_not", "okul_adi" field'lari olsun
- b) Tabloya 5 ogrenci ekleyin ortalamalari 2.6, 1.9, 3.2, 3.8, 3.5 olsun.
- c) Ortalamasi 3.0 ve ustu olan ogrencilerin statu field'ina "odullu ogrenci" yazdirin.



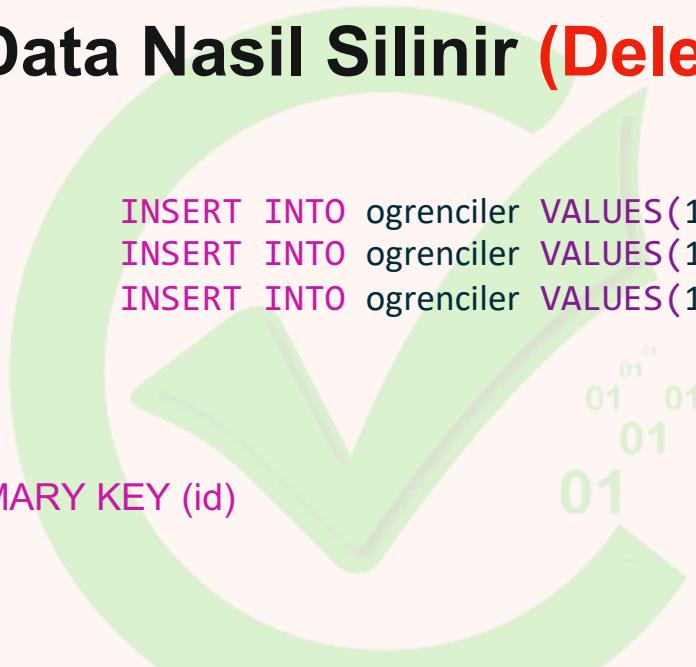
Practice 10:

- a) "Ogrenciler" isminde bir tablo olusturun,icinde "id", "isim", "ortalama_not", "okul_ismi" field'lari olsun.
- b) Tabloya 5 tane ogrenci ekleyin, ortalamalari 2.6, 1.9, 3.2, 3.8, 3.5 olsun,okul isimleri birbirinden farkli olsun.
- c) "veliler" isminde bir tablo daha olusturun, icinde "ogrenci_id", "veli_isim", "okul_isim" field'leri olsun
- d) 5 veli ekleyin, okul isimleri ogrenci tablosundaki 5 okul adi olsun, her velinin okulu farkli olsun.
- e) Okul adi ayni olan ogrenci ismi yerine veli ismini yazin

T E C H P R O E D

Tablodan Data Nasil Silinir (Delete)?

```
CREATE TABLE ogrenciler
(
    id char(3),
    isim varchar2(50),
    veli_isim varchar2(50),
    yazili_notu number(3),
    CONSTRAINT ogrenciler_pk PRIMARY KEY (id)
);
```



```
INSERT INTO ogrenciler VALUES(123, 'Ali Can', 'Hasan', 75);
INSERT INTO ogrenciler VALUES(124, 'Merve Gul', 'Ayse', 85);
INSERT INTO ogrenciler VALUES(125, 'Kemal Yasa', 'Hasan', 85);
```

- 1) “**DELETE FROM ogrenciler**” tablodaki tum datalari siler, fakat tabloyu silmez.
“**DELETE FROM ogrenciler**”, kodunu kullaninca bos bir tablo kalir.

T no data found H P R O E D

Tablodan Data Nasıl Silinir (Delete)?

2) CREATE TABLE ogrenciler
(
id char(3),
isim varchar2(50),
veli_isim varchar2(50),
yazili_notu number(3),
CONSTRAINT ogrenciler_pk
PRIMARY KEY (id)
);

INSERT INTO ogrenciler VALUES(123, 'Ali Can', 'Hasan', 75);
INSERT INTO ogrenciler VALUES(124, 'Merve Gul', 'Ayse', 85);
INSERT INTO ogrenciler VALUES(125, 'Kemal Yasa', 'Hasan', 85);

ID	ISIM	VELI_ISIM	YAZILI_NOTU
123	Ali Can	Hasan	75
124	Merve Gul	Ayse	85
125	Kemal Yasa	Hasan	85

“ DELETE FROM ogrenciler WHERE isim = ‘Ali Can’ ” kodu isim olarak Ali Can girilen kaydi (record) siler.

“ DELETE FROM ogrenciler WHERE yazili_notu = 85 ” kodu not olarak 85 girilen kaydi (record) siler.

ID	ISIM	VELI_ISIM	YAZILI_NOTU
124	Merve Gul	Ayse	85
125	Kemal Yasa	Hasan	85



Tablodan Data Nasıl Silinir (Delete)?

3) CREATE TABLE ogrenciler
(
id char(3),
isim varchar2(50),
veli_isim varchar2(50),
yazili_notu number(3),
CONSTRAINT ogrenciler_pk
PRIMARY KEY (id)
);

INSERT INTO ogrenciler VALUES(123, 'Ali Can', 'Hasan', 75);
INSERT INTO ogrenciler VALUES(124, 'Merve Gul', 'Ayse', 85);
INSERT INTO ogrenciler VALUES(125, 'Kemal Yasa', 'Hasan', 85);

ID	ISIM	VELI_ISIM	YAZILI_NOTU
123	Ali Can	Hasan	75
124	Merve Gul	Ayse	85
125	Kemal Yasa	Hasan	85

"DELETE FROM ogrenciler WHERE isim = 'Ali Can' OR veli_isim='Ayse'" kodu isim olarak Ali Can veya veli_isim olarak Ayse girilen kaydi (record) siler.

ID	ISIM	VELI_ISIM	YAZILI_NOTU
124	Merve Gul	Ayse	85
125	Kemal Yasa	Hasan	85



Practice 12:

ID	ISIM	SOYISIM	EMAIL	ISE_BASLAMA_TAR	IS_UNVANI	MAAS
123456789	Ali	Can	alican@gmail.com	10-APR-10	isci	5000
123456788	Veli	Cem	velicem@gmail.com	10-JAN-12	isci	5500
123456787	Ayse	Gul	aysegul@gmail.com	01-MAY-14	muhasebeci	4500
123456789	Fatma	Yasa	fatmayasa@gmail.com	10-APR-09	muhendis	7500

- 1) Verilen bilgilerin oldugu bir tablo olusturun
- 2) Tumisci'lerin maasina %20 zam yapin
- 3) Muhendis'lerin maasini 7000 yapin
- 4) Muhasebecinin adini 'Sena' soyadini 'Yilmaz' yapin
- 5) Maasi 5000 den buyuk olanlari silin

TECHPROED

Tablodan Data Nasil Silinir (**TRUNCATE**)?

- “**Truncate**” kodu kullanilarak bir tablodaki kayitlar silinirse datalarin geri getirilme ihtimali olmaz
- “**Truncate**” kodu geri getirilmesini (**rolling back**) istemeyeceginiz tablolari silmek icin kullanilir.

TRUNCATE TABLE customers

DELETE FROM customers;

INTERVIEW QUESTION : TRUNCATE, DELETE FROM VE DROP arasindaki fark nedir ?

T **L** **H** **P** **R** **U** **E** **D**
DELETE FROM ile sildigimiz kayitlari geri getirebiliriz ama TRUNCATE ile silinen kayitlar geri getirilemez.

Tablo Nasil Silinir (Drop)?

(Tum Tablo Icerigi ve Tablo Yapisi)

Ogrenciler isminde bir Tablo olusturun icinde id,isim,yazili_not,adres ve son degistirme field'lari olsun.
3 kisiyi tabloya ekleyin

```
CREATE TABLE ogrenciler  
(  
    id number(9),  
    isim varchar2(50),  
    adres varchar2(100),  
    yazili_not number(3)  
);
```

```
INSERT INTO ogrenciler VALUES(123, 'Ali Can', 'Ankara', 75);  
INSERT INTO ogrenciler VALUES(124, 'Merve Gul', 'Ankara', 85);  
INSERT INTO ogrenciler VALUES(125, 'Kemal Yasa', 'Istanbul', 85);
```

DROP TABLE ogrenciler; ogrenciler Tablosunu siler ve RecycleBin'e yollar
FLASHBACK TABLE ogrenciler TO BEFORE DROP; dosyayı geri alır.
PURGE TABLE ogrenciler; RecycleBin'de olan dosyayı geri getirilemeyecek şekilde siler
DROP TABLE ogrenciler PURGE; Kullandığımız bir dosyayı getirilemeyecek şekilde siler

UYARI: Purge kullandığımızda Tabloyu vedataları geri getirmek mümkün degildir.

Purge Kullanmanın Amacı: Hassas bilgileri silmek istediğinizde başka insanların o bilgiye ulaşamayacağından emin olursunuz

IN CONDITION

IN Condition birden fazla mantıksal ifade ile tanımlayabileceğimiz durumlari
(Condition) tek komutla yazabilme imkani verir

```
CREATE TABLE musteriler
(
    urun_id number(10),
    musteri_isim varchar2(50),
    urun_isim varchar2(50)
);
```

```
SELECT *
FROM musteriler
WHERE urun_isim ='Orange' OR urun_isim ='Apple' OR urun_isim ='Apricot';
```

```
SELECT *
FROM musteriler
WHERE urun_isim IN ('Orange', 'Apple', 'Apricot');
```

```
INSERT INTO musteriler VALUES (10, 'Mark', 'Orange');
INSERT INTO musteriler VALUES (10, 'Mark', 'Orange');
INSERT INTO musteriler VALUES (20, 'John', 'Apple');
INSERT INTO musteriler VALUES (30, 'Amy', 'Palm');
INSERT INTO musteriler VALUES (20, 'Mark', 'Apple');
INSERT INTO musteriler VALUES (10, 'Adem', 'Orange');
INSERT INTO musteriler VALUES (40, 'John', 'Apricot');
INSERT INTO musteriler VALUES (20, 'Eddie', 'Apple');
```

URUN_ID	MUSTERI_ISIM	URUN_ISIM
10	Mark	Orange
10	Mark	Orange
20	John	Apple
20	Mark	Apple
10	Adem	Orange
40	John	Apricot
20	Eddie	Apple

BETWEEN CONDITION

BETWEEN Condition iki mantiksal ifade ile tanimlayabilecegimiz durumlari tek komutla yazabilme imkani verir. Yazdigimiz 2 sinirda araliga dahildir (**INCLUSIVE**)

```
CREATE TABLE musteriler  
(  
    urun_id number(10),  
    musteri_isim varchar2(50),  
    urun_isim varchar2(50)  
)
```

```
INSERT INTO musteriler VALUES (10, 'Mark', 'Orange');  
INSERT INTO musteriler VALUES (10, 'Mark', 'Orange');  
INSERT INTO musteriler VALUES (20, 'John', 'Apple');  
INSERT INTO musteriler VALUES (30, 'Amy', 'Palm');  
INSERT INTO musteriler VALUES (20, 'Mark', 'Apple');  
INSERT INTO musteriler VALUES (10, 'Adem', 'Orange');  
INSERT INTO musteriler VALUES (40, 'John', 'Apricot');  
INSERT INTO musteriler VALUES (20, 'Eddie', 'Apple');
```

1) Urun_id 20 ile 40 arasında olan ürünlerin tüm bilgilerini listeleyiniz

```
SELECT *  
FROM musteriler  
WHERE urun_id>=20 AND urun_id<=40;
```

```
SELECT *  
FROM musteriler  
WHERE urun_id BETWEEN 20 AND 40 ;
```

URUN_ID	MUSTERI_ISIM	URUN_ISIM
20	John	Apple
30	Amy	Palm
20	Mark	Apple
40	John	Apricot
20	Eddie	Apple

2) İsminin ilk harfi E ile J arasında olan kişilerin tüm bilgilerini listeleyin

NOT BETWEEN CONDITION

NOT BETWEEN Condition iki mantiksal ifade ile tanimlayabilecegimiz durumlari tek komutla yazabilme imkani verir. Yazdigimiz 2 sinirda araliga harictir (**EXCLUSIVE**)

```
CREATE TABLE musteriler  
(  
    urun_id number(10),  
    musteri_isim varchar2(50),  
    urun_isim varchar2(50)  
);
```

```
INSERT INTO musteriler VALUES (10, 'Mark', 'Orange');  
INSERT INTO musteriler VALUES (10, 'Mark', 'Orange');  
INSERT INTO musteriler VALUES (20, 'John', 'Apple');  
INSERT INTO musteriler VALUES (30, 'Amy', 'Palm');  
INSERT INTO musteriler VALUES (20, 'Mark', 'Apple');  
INSERT INTO musteriler VALUES (10, 'Adem', 'Orange');  
INSERT INTO musteriler VALUES (40, 'John', 'Apricot');  
INSERT INTO musteriler VALUES (20, 'Eddie', 'Apple');
```

1) Urun_id 20 ile 40 arasında olmayan ürünlerin tüm bilgilerini listeleyiniz

```
SELECT *  
FROM musteriler  
WHERE urun_id<20 OR urun_id>30;
```

```
SELECT *  
FROM musteriler  
WHERE urun_id NOT BETWEEN 20 AND 40 ;
```

2) İsminin ilk harfi E ile J arasında olmayan kişilerin tüm bilgilerini listeleyin

URUN_ID	MUSTERI_ISIM	URUN_ISIM
10	Mark	Orange
10	Mark	Orange
10	Adem	Orange
40	John	Apricot

Practice 13

ID	ISIM	SOYISIM	EMAIL	ISE_BASLAMA_TAR	IS_UNVANI	MAAS
123456789	Ali	Can	alican@gmail.com	10-APR-10	isci	5000
123456788	Veli	Cem	velicem@gmail.com	10-JAN-12	isci	5500
123456787	Ayse	Gul	aysegul@gmail.com	01-MAY-14	muhasebeci	4500
123456789	Fatma	Yasa	fatmayasa@gmail.com	10-APR-09	muhendis	7500

- a) Yukarda verilen “personel” tablosunu olusturun
- b) Tablodan maasi 5000’den az veya unvaniisci olanların isimlerini listeleyin
- c) Iscilerin tum bilgilerini listeleyin
- d) Soyadi Can,Cem veya Gul olanların unvanlarını ve maaslarını listeleyin
- e) Maasi 5000’den cok olanların emailve is baslama tarihlerini listeleyin
- f) Maasi 5000’den cok veya 7000’den az olanların tum bilgilerini listeleyin

TECHPROED

Tekrar Soruları

- 1) DELETE ile TRUNCATE arasındaki fark nedir ?
- 2) DELETE ile DROP arasındaki fark nedir?
- 3) DROP ile DROP PURGE arasındaki fark nedir?
- 4) Aşağıdaki sorgu ile aynı sonucu veren bir sorgu yazınız.

```
SELECT *  
FROM ogrenciler  
WHERE yas>=8 AND yas<=17;
```

- 5) Aşağıdaki sorgu ile aynı sonucu veren bir sorgu yazınız

```
SELECT *  
FROM ogrenciler  
WHERE yas<8 OR yas>17;
```

- 6) Aşağıdaki sorgu ile aynı sonucu veren bir sorgu yazınız

```
SELECT *  
FROM ogrenciler  
WHERE yas = 6 OR yas = 7 OR yas = 8 OR yas = 9;
```

KISA TEKRAR

Personel isminde bir tablo olusturun. Icinde id, isim, sehir, maas ve sirket field'lari olsun. Id'yi 2.yontemle PK yapin

```
CREATE TABLE personel
(
    id number(9),
    isim varchar2(50),
    sehir varchar2(50),
    maas number(20),
    sirket varchar2(20),
    CONSTRAINT personel_pk PRIMARY KEY (id)
);
```

Personel_bilgi isminde bir tablo olusturun. Icinde id, tel ve cocuk sayisi field'lari olsun. Id'yi FK yapin ve personel tablosu ile relation kurun

```
CREATE TABLE personel_bilgi
(
    id number(9),
    tel char(10) UNIQUE ,
    cocuk_sayisi number(2),
    CONSTRAINT personel_bilgi_fk FOREIGN KEY (id) REFERENCES personel(id)
);
```

```
INSERT INTO personel VALUES(123456789, 'Ali Yilmaz', 'Istanbul', 5500, 'Honda');
INSERT INTO personel VALUES(234567890, 'Veli Sahin', 'Istanbul', 4500, 'Toyota');
INSERT INTO personel VALUES(345678901, 'Mehmet Ozturk', 'Ankara', 3500, 'Honda');
INSERT INTO personel VALUES(456789012, 'Mehmet Ozturk', 'Izmir', 6000, 'Ford');
INSERT INTO personel VALUES(567890123, 'Mehmet Ozturk', 'Ankara', 7000, 'Tofas');
INSERT INTO personel VALUES(456715012, 'Veli Sahin', 'Ankara', 4500, 'Ford');
INSERT INTO personel VALUES(123456710, 'Hatice Sahin', 'Bursa', 4500, 'Honda');
```

01

```
INSERT INTO personel_bilgi VALUES(123456789, '5302345678' , 5);
INSERT INTO personel_bilgi VALUES(234567890, '5422345678' , 4);
INSERT INTO personel_bilgi VALUES(345678901, '5354561245' , 3);
INSERT INTO personel_bilgi VALUES(456789012, '5411452659' , 3);
INSERT INTO personel_bilgi VALUES(567890123, '5551253698' , 2);
INSERT INTO personel_bilgi VALUES(456789012, '5524578574' , 2);
INSERT INTO personel_bilgi VALUES(123456710, '5537488585' , 1);
```

KISA TEKRAR

ID	ISIM	SEHIR	MAAS	SIRKET
123456789	Ali Yilmaz	Istanbul	5500	Honda
234567890	Veli Sahin	Istanbul	4500	Toyota
345678901	Mehmet Ozturk	Ankara	3500	Honda
456789012	Mehmet Ozturk	Izmir	6000	Ford
567890123	Mehmet Ozturk	Ankara	7000	Tofas
456789152	Veli Sahin	Ankara	4500	Ford
123456710	Hatice Sahin	Bursa	4500	Honda

ID	TEL	COCUK SAYISI
123456789	5302345678	5
234567890	5422345678	4
345678901	5354561245	3
456789012	5411452659	3
567890123	5551253698	2
456789012	5524578574	2
123456710	5537488585	1

SORU 1) Personel_bilgi tablosundan 5 cocugu olan kisinin cocuk sayisini 2 yapin

```
UPDATE personel_bilgi  
SET cocuk_sayisi=2  
WHERE cocuk_sayisi=5;
```

TECH PRO EDITION

Tekrar Soruları

SORU 2) Personel tablosundan ücreti 4500 veya 5000 olanların maaşlarını %10 artırın

```
UPDATE personel  
SET maas=maas*1.1  
WHERE maas IN (4500,5000);
```

SORU 3) Personel tablosundan maaşı 4950 olanları silin

```
DELETE FROM personel  
WHERE maas =4900;
```

ORA-02292: integrity constraint (SQL_AHZDXVGZXDBVBVLYOPIBABNDG.PERSONEL_BILGI_FK) violated - child record found ORA-06512: at "SYS.DBMS_SQL", line 1721

Tekrar Soruları

SORU 4) Personel_bilgi tablosundan 3 veya 4 cocugu olanlari silin

```
DELETE FROM personel_bilgi  
WHERE cocuk_sayisi IN (3,4);
```

SORU 5) Personel tablosundan HONDA'da calisip maasi 3500 olanlari silin

```
DELETE FROM personel  
WHERE maas =3500 AND sirket='Honda';
```

T E C H P R O E D

Tekrar Soruları

SORU 6) Personel_bilgi tablosundan dataları geri getirilemeyecek şekilde silin

TRUNCATE TABLE personel_bilgi;

SORU 7) Personel tablosundan maasi 4000 ile 5000 arasında olanları silin

DELETE FROM personel
WHERE maas BETWEEN 4000 AND 5000;

T E C H P R O E D

Tekrar Soruları

SORU 8) Personel tablosundan maasi 5000 ile 6000 arasında olmayanları silin

```
DELETE FROM personel  
WHERE maas NOT BETWEEN 5000 AND 6000;
```

SORU 9) Personel tablosunu geri getirilemeyecek şekilde silin

```
DROP TABLE personel PURGE; HATA VERIR
```

Once personel_bilgi tablosunu silin

```
DROP TABLE personel_bilgi;
```

Personel tablosunu geri getirilemeyecek şekilde silin

SUBQUERIES

SUBQUERY baska bir SORGU(query)'nun icinde calisan SORGU'dur.

- 1) **WHERE'** den sonra kullanilabilir

```
CREATE TABLE personel
(
    id number(9),
    isim varchar2(50),
    sehir varchar2(50),
    maas number(20),
    sirket varchar2(20)
);
```

```
CREATE TABLE sirketler
(
    sirket_id number(9),
    sirket varchar2(20),
    personel_sayisi number(20)
);
```

```
INSERT INTO personel2 VALUES(123456789, 'Ali Seker', 'Istanbul', 2500, 'Honda');
INSERT INTO personel2 VALUES(234567890, 'Ayse Gul', 'Istanbul', 1500, 'Toyota');
INSERT INTO personel2 VALUES(345678901, 'Veli Yilmaz', 'Ankara', 3000, 'Honda');
INSERT INTO personel2 VALUES(456789012, 'Veli Yilmaz', 'Izmir', 1000, 'Ford');
INSERT INTO personel2 VALUES(567890123, 'Veli Yilmaz', 'Ankara', 7000, 'Hyundai');
INSERT INTO personel2 VALUES(456789012, 'Ayse Gul', 'Ankara', 1500, 'Ford');
INSERT INTO personel2 VALUES(123456710, 'Fatma Yasa', 'Bursa', 2500, 'Honda');
```

```
INSERT INTO sirketler VALUES(100, 'Honda', 12000);
INSERT INTO sirketler VALUES(101, 'Ford', 18000);
INSERT INTO sirketler VALUES(102, 'Hyundai', 10000);
INSERT INTO sirketler VALUES(103, 'Toyota', 21000);
```

SUBQUERIES

1) Personel sayisi 15.000'den cok olan sirketlerin isimlerini ve bu sirkette calisan personelin isimlerini listeleyin

```
SELECT isim, sirket  
FROM personel  
WHERE sirket IN ( SELECT sirket  
                  FROM sirketler  
                  WHERE personel_sayisi > 15000);
```

SIRKET
Ford
Toyota

Subquery sonuçu

ISIM	SIRKET
Ayse Gul	Toyota
Veli Yilmaz	Ford
Ayse Gul	Ford

Query sonuçu

2) Sirket_id'si 101'den buyuk olan sirketlerin verdikleri maaslari ve o maasi alanların sehirlerini listeleyiniz

```
SELECT sehir, maas  
FROM personel  
WHERE sirket IN (SELECT sirket  
                  FROM sirketler  
                  WHERE sirket_id > 101);
```

SIRKET
Hyundai
Toyota

Subquery sonuçu

SEHIR	MAAS
Ankara	7000
Istanbul	1500

Query sonuçu

SIRKET	PERSONEL_SAYISI
Honda	12000
Hyundai	10000
Ford	18000

3 -- Ankara'da personeli olan sirketlerin sirket id ve calisan sayilarini listeleyiniz

SUBQUERIES

- 2) **SELECT**' den sonra kullanılır.

Ancak SELECT CLAUSE da kullanılan Subquery **SADECE 1 DEGER** donmelidir.

Dolayısıyla SUM, COUNT, MIN, MAX ve AVG gibi fonksiyonlar kullanılır.

Bu fonksiyonlara **AGGREGATE FUNCTION** denir.

SORU 1- Her sirketin ismini, personel sayisini ve personelin ortalama maasini listeleyen bir QUERY yazın.

SELECT sirket,personel sayisi,(**SELECT AVG(maas)**

FROM personnel

WHERE sirketler.sirket=personel.sirket

)

FROM sirketler;

SUBQUERIES

SORU 2- Her sirketin ismini ve personelin aldigı max. maasi listeleyen bir QUERY yazın.

```
SELECT sirket,personel_sayisi,(SELECT MAX(maas)
    FROM personel
    WHERE sirketler.sirket=personel.sirket
) AS sirketteki_Max_Maas,
FROM sirketler;
```

SIRKET	PERSONEL SAYISI	SIRKETTEKİ_MAX_MAAS
Honda	12000	3000
Ford	18000	1500
Hyundai	10000	7000
Toyota	21000	1500

TECHP

D

SUBQUERIES

SORU 3- Her sirketin id'sini, ismini ve toplam kac sehirde bulundugunu listeleyen bir QUERY yaziniz.

```
SELECT sirket_id,sirket,(SELECT COUNT(sehir)
                           FROM personel
                          WHERE sirketler.sirket=personel.sirket
                        ) bulundugu_sehir_sayisi
      FROM sirketler;
```

SIRKET	PERSONEL SAYISI	BULUNDUGU_SEHIR SAYISI
Honda	12000	3
Ford	18000	2
Hyundai	10000	1
Toyota	21000	1

TECH

ED

SUBQUERIES

SORU 4- Her sirketin ismini, personel sayisini ve personelin aldigı max. ve min. maasi listeleyen bir QUERY yazın.

```
SELECT sirket,personel_sayisi,(SELECT MAX(maas)
    FROM personel
    WHERE sirketler.sirket=personel.sirket
) AS max_maas,
    (SELECT MIN(maas)
    FROM personel
    WHERE sirketler.sirket=personel.sirket
) AS min_maas
FROM sirketler;
```

SIRKET	PERSONEL SAYISI	MAX_MAAS	MIN_MAAS
Honda	12000	3000	2500
Ford	18000	1500	1000
Hyundai	10000	7000	7000
Toyota	21000	1500	1500

T E C H P I

SUBQUERIES

SORU 5- Her sirketin ismini, personel sayisini ve tablodaki iscilerine odedigi toplam maasi listeleyen bir QUERY yazin.

```
SELECT sirket,personel_sayisi,(SELECT SUM(maas)
                                FROM personel
                               WHERE sirketler.sirket=personel.sirket
                             ) AS toplam_maas
   FROM sirketler;
```

SIRKET	PERSONEL SAYISI	TOPLAM MAAS
Honda	12000	8000
Ford	18000	2500
Hyundai	10000	7000
Toyota	21000	1500

T E C H P R O E D

EXISTS CONDITION

EXISTS Condition subquery'ler ile kullanılır. IN ifadesinin kullanımına benzer olarak, EXISTS ve NOT EXISTS ifadeleri de alt sorgudan getirilen değerlerin içerisinde bir değerin olması veya olmaması durumunda işlem yapılmasını sağlar.

```
CREATE TABLE mart_satislar
(
    urun_id number(10),
    musteri_isim varchar2(50),
    urun_isim varchar2(50)
);
```

```
CREATE TABLE nisan_satislar
(
    urun_id number(10),
    musteri_isim varchar2(50),
    urun_isim varchar2(50)
);
```

```
INSERT INTO mart_satislar VALUES (10, 'Mark', 'Honda');
INSERT INTO mart_satislar VALUES (10, 'Mark', 'Honda');
INSERT INTO mart_satislar VALUES (20, 'John', 'Toyota');
INSERT INTO mart_satislar VALUES (30, 'Amy', 'Ford');
INSERT INTO mart_satislar VALUES (20, 'Mark', 'Toyota');
INSERT INTO mart_satislar VALUES (10, 'Adem', 'Honda');
INSERT INTO mart_satislar VALUES (40, 'John', 'Hyundai');
INSERT INTO mart_satislar VALUES (20, 'Eddie', 'Toyota');
```

```
INSERT INTO nisan_satislar VALUES (10, 'Hasan', 'Honda');
INSERT INTO nisan_satislar VALUES (10, 'Kemal', 'Honda');
INSERT INTO nisan_satislar VALUES (20, 'Ayse', 'Toyota');
INSERT INTO nisan_satislar VALUES (50, 'Yasar', 'Volvo');
INSERT INTO nisan_satislar VALUES (20, 'Mine', 'Toyota');
```

EXISTS CONDITION

Her iki ayda da aynı id ile satılan ürünlerin urun_id'lerini ve ürünlerin mart ayında alanların isimlerini getiren bir query yazınız..

URUN_ID	MUSTERI_ISIM	URUN_ISIM
10	Mark	Honda
10	Mark	Honda
20	John	Toyota
30	Amy	Ford
20	Mark	Toyota
10	Adem	Honda
40	John	Hyundai
20	Eddie	Toyota

URUN_ID	MUSTERI_ISIM	URUN_ISIM
10	Hasan	Honda
10	Kemal	Honda
20	Ayse	Toyota
50	Yasar	Volvo
20	Mine	Toyota

URUN_ID	MUSTERI_ISIM
10	Mark
10	Mark
10	Adem
20	John
20	Mark
20	Eddie

SELECT musteri_isim
FROM mart_satislar
WHERE EXISTS (SELECT urun_id
FROM nisan_satislar
WHERE mart_satislar.urun_id = nisan_satislar.urun_id);

EXISTS CONDITION

Her iki ayda da satilan urun_isimleri ayni urunlerin urun_isim'ini ve urunleri nisan ayinda alanların isimlerini getiren bir query yaziniz..

```
SELECT urun_isim, musteri_isim  
FROM nisan_satislar  
WHERE EXISTS (SELECT urun_isim  
               FROM mart_satislar  
               WHERE mart_satislar.urun_isim = nisan_satislar.urun_isim);
```

URUN_ISIM	MUSTERI_ISIM
Honda	Hasan
Honda	Kemal
Toyota	Ayse
Toyota	Mine

```
SELECT musteri_isim  
FROM nisan_satislar  
WHERE NOT EXISTS (SELECT urun_isim  
                   FROM mart_satislar  
                   WHERE mart_satislar.urun_isim = nisan_satislar.urun_isim);
```

URUN_ISIM	MUSTERI_ISIM
Volvo	Yasar

IS NULL CONDITION

Arama yapılan field'da NULL değeri almış kayıtları getirir.

```
CREATE TABLE insanlar
(
    ssn char(9),
    isim varchar2(50),
    adres varchar2(50)
);
```

```
SELECT *
FROM insanlar
WHERE isim IS NULL;
```

SSN	NAME	ADDRESS
456789012	-	Bursa
567890123	-	Denizli

```
INSERT INTO insanlar VALUES(123456789, 'Ali Can', 'Istanbul');
INSERT INTO insanlar VALUES(234567890, 'Veli Cem', 'Ankara');
INSERT INTO insanlar VALUES(345678901, 'Mine Bulut', 'Izmir');
INSERT INTO insanlar (ssn, adres) VALUES(456789012, 'Bursa');
INSERT INTO insanlar (ssn, adres) VALUES(567890123, 'Denizli');
```

SSN	NAME	ADDRESS
123456789	Ali Can	Istanbul
234567890	Veli Cem	Ankara
345678901	Mine Bulut	Izmir
456789012	-	Bursa
567890123	-	Denizli

```
01
SELECT *
FROM insanlar
WHERE isim IS NOT NULL;
```

```
UPDATE insanlar
SET isim = 'Isim Girilmemis'
WHERE name IS NULL;
```

SSN	NAME	ADDRESS
456789012	Isim Girilmemis	Bursa
567890123	Isim Girilmemis	Denizli

SSN	NAME	ADDRESS
123456789	Ali Can	-
234567890	Veli Cem	Ankara
345678901	Mine Bulut	Izmir

ORDER BY CLAUSE

ORDER BY komutu belli bir field'a gore NATURAL ORDER olarak siralama yapmak icin kullanilir

ORDER BY komutu sadece SELECT komutu ile kullanilir

CREATE TABLE insanlar

```
(  
    ssn char(9),  
    isim varchar2(50),  
    soyisim varchar2(50),  
    adres varchar2(50)  
)
```

Insanlar tablosundaki datalari adres'e gore siralayin

```
SELECT *  
FROM insanlar  
ORDER BY adres;
```

```
INSERT INTO insanlar VALUES(123456789, 'Ali', 'Can', 'Istanbul');  
INSERT INTO insanlar VALUES(234567890, 'Veli', 'Cem', 'Ankara');  
INSERT INTO insanlar VALUES(345678901, 'Mine', 'Bulut', 'Ankara');  
INSERT INTO insanlar VALUES(256789012, 'Mahmut', 'Bulut', 'Istanbul ');  
INSERT INTO insanlar VALUES (344678901, 'Mine', 'Yasa', 'Ankara');  
INSERT INTO insanlar VALUES (345678901, 'Veli', 'Yilmaz', 'Istanbul ');
```

SSN	ISIM	SOYISIM	ADRES	SSN	ISIM	SOYISIM	ADRES
123456789	Ali	Can	Istanbul	345678901	Mine	Bulut	Ankara
234567890	Veli	Cem	Ankara	344678901	Mine	Yasa	Ankara
345678901	Mine	Bulut	Ankara	234567890	Veli	Cem	Ankara
256789012	Mahmut	Bulut	Istanbul	123456789	Ali	Can	Istanbul
344678901	Mine	Yasa	Ankara	345678901	Veli	Yilmaz	Istanbul
345678901	Veli	Yilmaz	Istanbul	256789012	Mahmut	Bulut	Istanbul

ORDER BY CLAUSE

İnsanlar tablosundaki ismi Mine olanları SSN sıralı olarak listeleyin

```
SELECT *
FROM insanlar
WHERE isim='Mine'
ORDER BY ssn;
```

SSN	ISIM	SOYISIM	ADRES
344678901	Mine	Yasa	Ankara
345678901	Mine	Bulut	Ankara

NOT : Order By komutundan sonra field ismi yerine field numarası da kullanılabilir

İnsanlar tablosundaki soyismi Bulut olanları isim sıralı olarak listeleyin

```
SELECT *
FROM insanlar
WHERE soyisim='Bulut'
ORDER BY 2;
```

SSN	ISIM	SOYISIM	ADRES
256789012	Mahmut	Bulut	Istanbul
345678901	Mine	Bulut	Ankara

ORDER BY field_name DESC CLAUSE

Insanlar tablosundaki tum kayitlari SSN numarasi buyukten kucuge olarak siralayin

```
SELECT *
FROM insanlar
ORDER BY ssn DESC;
```

SSN	ISIM	SOYISIM	ADRES
345678901	Mine	Bulut	Ankara
345678901	Veli	Yilmaz	Istanbul
344678901	Mine	Yasa	Ankara
256789012	Mahmut	Bulut	Istanbul
234567890	Veli	Cem	Ankara
123456789	Ali	Can	Istanbul

Insanlar tablosundaki tum kayitlari isimler Natural sirali, Soyisimler ters sirali olarak listeleyin

```
SELECT *
FROM insanlar
ORDER BY isim ASC, soyisim DESC;
```

SSN	ISIM	SOYISIM	ADRES
123456789	Ali	Can	Istanbul
256789012	Mahmut	Bulut	Istanbul
344678901	Mine	Yasa	Ankara
345678901	Mine	Bulut	Ankara
345678901	Veli	Yilmaz	Istanbul
234567890	Veli	Cem	Ankara

ALIASES

Aliases kodu ile tablo yazdirilirken, field isimleri sadece o cikti icin degistirilebilir

```
CREATE TABLE calisanlar  
(  
    calisan_id char(9),  
    calisan_isim varchar2(50),  
    calisan_dogdugu_sehir varchar2(50)  
);
```

```
SELECT calisan_id AS id, calisan_isim AS isim, calisan_dogdugu_sehir AS dogum_yeri  
FROM calisanlar;
```

```
SELECT calisan_id AS id, calisan_isim || calisan_dogdugu_sehir AS isim_ve_dogum_yeri  
FROM calisanlar;
```

```
INSERT INTO calisanlar VALUES(123456789, 'Ali Can', 'Istanbul');  
INSERT INTO calisanlar VALUES(234567890, 'Veli Cem', 'Ankara');  
INSERT INTO calisanlar VALUES(345678901, 'Mine Bulut', 'Izmir');
```

EMPLOYEE_ID	EMPLOYEE_NAME	EMPLOYEE_BIRTH_CITY
123456789	Ali Can	Istanbul
234567890	Veli Cem	Ankara
345678901	Mine Bulut	Izmir

ID	ISIM	DOGUM_YERI
123456789	Ali Can	Istanbul
234567890	Veli Cem	Ankara
345678901	Mine Bulut	Izmir

ID	ISIM_VE_DOGUM_YERI
123456789	Ali CanIstanbul
234567890	Veli CemAnkara
345678901	Mine BulutIzmir

TECHPRO

GROUP BY CLAUSE

Group By komutu sonuçları bir veya daha fazla sütuna göre grüplamak için **SELECT** komutuyla birlikte kullanılır

```
CREATE TABLE manav
(
    isim varchar2(50),
    Urun_adi varchar2(50),
    Urun_miktar number(9)
);
```

```
INSERT INTO manav VALUES('Ali', 'Elma', 5);
INSERT INTO manav VALUES('Ayse', 'Armut', 3);
INSERT INTO manav VALUES('Veli', 'Elma', 2);
INSERT INTO manav VALUES('Hasan', 'Uzum', 4);
INSERT INTO manav VALUES('Ali', 'Armut', 2);
INSERT INTO manav VALUES('Ayse', 'Elma', 3);
INSERT INTO manav VALUES('Veli', 'Uzum', 5);
INSERT INTO manav VALUES('Ali', 'Armut', 2);
INSERT INTO manav VALUES('Veli', 'Elma', 3);
INSERT INTO manav VALUES('Ayse', 'Uzum', 2);
```

ISIM	URUN_ADI	URUN_MIKTAR
Ali	Elma	5
Ayse	Armut	3
Veli	Elma	2
Hasan	Uzum	4
Ali	Armut	2
Ayse	Elma	3
Veli	Uzum	5
Ali	Armut	2
Veli	Elma	3
Ayse	Uzum	2

1) Isme göre alınan toplam ürünler bulun

```
SELECT isim, SUM(urun_miktar) AS Alinan_Toplama_Meyve
FROM manav
GROUP BY isim;
```

ISIM	ALINAN_TOPLAM_MEYVE
Veli	10
Ayse	8
Ali	9
Hasan	4

GROUP BY CLAUSE

ISIM	URUN_ADI	URUN_MIKTAR
Ali	Elma	5
Ayse	Armut	3
Veli	Elma	2
Hasan	Uzum	4
Ali	Armut	2
Ayse	Elma	3
Veli	Uzum	5
Ali	Armut	2
Veli	Elma	3
Ayse	Uzum	2

2) Urun ismine gore urunu alan toplam kisi sayisi

```
SELECT urun_adi, COUNT(isim) AS UrunuAlanKisiSayisi  
FROM manav  
GROUP BY urun_adi;
```

URUN_ADI	URUNU_ALAN_KISI_SAYISI
Elma	4
Uzum	3
Armut	3

3) Alinan kilo miktarina gore musteri sayisi

```
SELECT urun_miktar, COUNT(isim) AS UrunMiktariniAlanKisiSayisi  
FROM manav  
GROUP BY urun_miktar;
```

URUN_MIKTAR	URUN_MIKTARINI_ALAN_KISI_SAYISI
2	4
5	2
4	1
3	3

T E C H P R O E D

GROUP BY CLAUSE

CREATE TABLE personel

```
(  
    id number(9),  
    isim varchar2(50),  
    sehir varchar2(50),  
    maas number(20),  
    sirket varchar2(20)  
)
```

1) Isme gore toplam maaslari bulun

```
SELECT isim, SUM(maas) AS toplam_maas  
FROM personel  
GROUP BY isim;
```

2) sehere gore toplam personel sayisini bulun

```
SELECT sehir, COUNT(isim) AS calisan_sayisi  
FROM personel  
GROUP BY sehir;
```

```
INSERT INTO personel VALUES(123456789, 'Ali Yilmaz', 'Istanbul', 5500, 'Honda');  
INSERT INTO personel VALUES(234567890, 'Veli Sahin', 'Istanbul', 4500, 'Toyota');  
INSERT INTO personel VALUES(345678901, 'Mehmet Ozturk', 'Ankara', 3500, 'Honda');  
INSERT INTO personel VALUES(456789012, 'Mehmet Ozturk', 'Izmir', 6000, 'Ford');  
INSERT INTO personel VALUES(567890123, 'Mehmet Ozturk', 'Ankara', 7000, 'Tofas');  
INSERT INTO personel VALUES(456789012, 'Veli Sahin', 'Ankara', 4500, 'Ford');  
INSERT INTO personel VALUES(123456710, 'Hatice Sahin', 'Bursa', 4500, 'Honda');
```

ISIM	TOPLAM_MAAS
Hatice Sahin	4500
Veli Sahin	9000
Ali Yilmaz	5500
Mehmet Ozturk	16500

SEHIR	CALISAN_SAYISI
Izmir	1
Bursa	1
Istanbul	2
Ankara	3

ID	ISIM	SEHIR	MAAS	SIRKET
123456789	Ali Yilmaz	Istanbul	5500	Honda
234567890	Veli Sahin	Istanbul	4500	Toyota
345678901	Mehmet Ozturk	Ankara	3500	Honda
456789012	Mehmet Ozturk	Izmir	6000	Ford
567890123	Mehmet Ozturk	Ankara	7000	Tofas
456789012	Veli Sahin	Ankara	4500	Ford
123456710	Hatice Sahin	Bursa	4500	Honda

GROUP BY CLAUSE

3) Sirketlere gore maasi 5000 liradan fazla olan personel sayisini bulun

```
SELECT sirket, COUNT (*) AS calisan_sayisi  
FROM personel  
WHERE maas>5000  
GROUP BY sirket;
```

SIRKET	CALISAN SAYISI
Honda	1
Ford	1
Tofas	1

4) Her sirket icin Min ve Max maasi bulun

```
SELECT sirket, MIN (maas) AS en_az_maas, MAX (maas) AS en_fazla_maas  
FROM personel  
GROUP BY sirket;
```

SIRKET	EN_AZ_MAAS	EN_FAZLA_MAAS
Honda	3500	5500
Ford	4500	6000
Toyota	4500	4500
Tofas	7000	7000

TECHPROED

HAVING CLAUSE

HAVING, AGGREGATE FUNCTION'lar ile birlikte kullanılan FILTRELEME komutudur.

```
CREATE TABLE personel  
(  
    id number(9),  
    isim varchar2(50),  
    sehir varchar2(50),  
    maas number(20),  
    sirket varchar2(20)  
);
```

```
INSERT INTO personel VALUES(123456789, 'Ali Yilmaz', 'Istanbul', 5500, 'Honda');  
INSERT INTO personel VALUES(234567890, 'Veli Sahin', 'Istanbul', 4500, 'Toyota');  
INSERT INTO personel VALUES(345678901, 'Mehmet Ozturk', 'Ankara', 3500, 'Honda');  
INSERT INTO personel VALUES(456789012, ' Mehmet Ozturk ', 'Izmir', 6000, 'Ford');  
INSERT INTO personel VALUES(567890123, ' Mehmet Ozturk ', 'Ankara', 7000, 'Tofas');  
INSERT INTO personel VALUES(456789012, ' Veli Sahin ', 'Ankara', 4500, 'Ford');  
INSERT INTO personel VALUES(123456710, 'Hatice Sahin', 'Bursa', 4500, 'Honda');
```

1) Her sirketin **MIN** maaslarini eger 2000'den buyukse goster

```
SELECT sirket, MIN (maas) AS en_az_maas  
FROM personel  
GROUP BY sirket  
HAVING MIN (maas) >2000;
```

SIRKET	EN_AZ_MAAS
Honda	3500
Ford	4500
Toyota	4500
Tofas	7000

HAVING CLAUSE

2) Ayni isimdeki kisilerin aldigı toplam gelir 10000 liradan fazla ise ismi ve toplam maasi gösteren soru yazınız

```
SELECT isim, SUM (maas) AS toplam_maas  
FROM personel  
GROUP BY isim  
HAVING SUM (maas) >10000;
```

ISIM	TOPLAM_MAAAS
Mehmet Ozturk	16500

3) Eger bir sehirde calisan personel sayisi 1'den coksa sehir ismini ve personel sayisini veren soru yazınız

```
SELECT sehir, COUNT (isim) AS toplam_personel_sayisi  
FROM personel  
GROUP BY sehir  
HAVING COUNT (isim) >1;
```

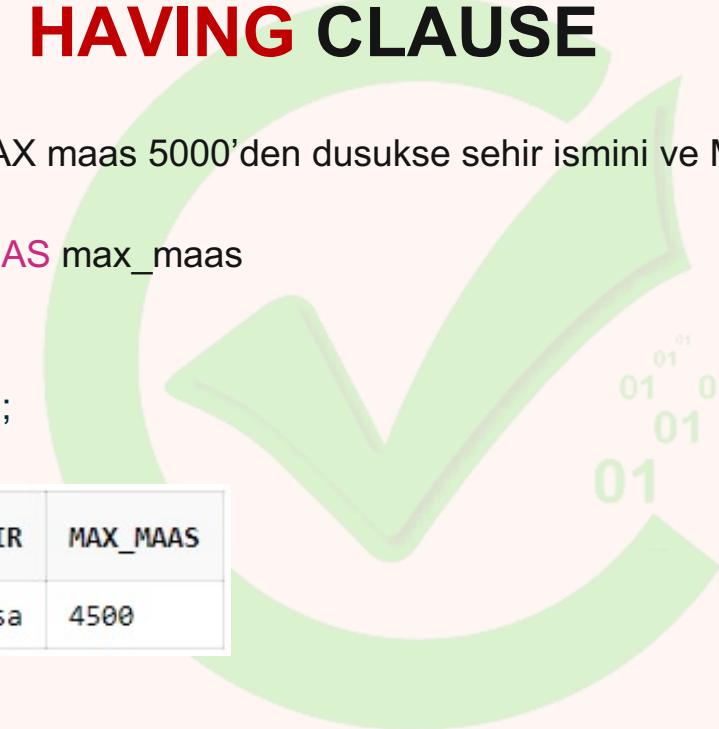
SEHIR	TOPLAM_PERSONEL_SAYISI
Istanbul	2
Ankara	3

HAVING CLAUSE

4) Eger bir sehirde alınan MAX maas 5000'den dusukse sehir ismini ve MAX maasi veren sorgu yaziniz

```
SELECT sehir, MAX (maas) AS max_maas  
FROM personel  
GROUP BY sehir  
HAVING MAX (maas) <5000;
```

SEHIR	MAX_MAAS
Bursa	4500



T E C H P R O E D

UNION OPERATOR

İki farklı sorgulamanın sonucunu birlestiren işlemidir. Secilen **Field SAYISI** ve **DATA TYPE**'i aynı olmalıdır.

```
CREATE TABLE personel
(
    id number(9),
    isim varchar2(50),
    sehir varchar2(50),
    maas number(20),
    sirket varchar2(20)
);
```

```
INSERT INTO personel VALUES(123456789, 'Ali Yilmaz', 'Istanbul', 5500, 'Honda');
INSERT INTO personel VALUES(234567890, 'Veli Sahin', 'Istanbul', 4500, 'Toyota');
INSERT INTO personel VALUES(345678901, 'Mehmet Ozturk', 'Ankara', 3500, 'Honda');
INSERT INTO personel VALUES(456789012, 'Mehmet Ozturk', 'Izmir', 6000, 'Ford');
INSERT INTO personel VALUES(567890123, 'Mehmet Ozturk', 'Ankara', 7000, 'Tofas');
INSERT INTO personel VALUES(456789012, 'Veli Sahin ', 'Ankara', 4500, 'Ford');
INSERT INTO personel VALUES(123456710, 'Hatice Sahin', 'Bursa', 4500, 'Honda');
```

1) Maasi 4000'den çok olan işçi isimlerini ve 5000 liradan fazla maas alinan sehirleri gosteren sorguyu yaziniz

```
SELECT sehir ASisci_veya_sehir_ismi ,maas
FROM personel
WHERE maas >5000
UNION
SELECT isim ASisci_veya_sehir_ismi , maas
FROM personel
WHERE maas > 4000;
```

ISCI_VEYA_SEHIR_ISMI	MAAS
Ali Yilmaz	5500
Ankara	4500
Ankara	7000
Bursa	4500
Hatice Sahin	4500
Istanbul	4500
Istanbul	5500
Izmir	6000
Mehmet Ozturk	6000
Mehmet Ozturk	7000
Veli Sahin	4500
Veli Sahin	4500

UNION OPERATOR

2) Mehmet Ozturk ismindeki kisilerin aldiği maaşları ve İstanbul'daki personelin maaşlarını bir tabloda gösteren sorgu yazınız

```
SELECT sehir ASisci_veya_sehir_ismi ,maas  
FROM personel  
WHERE sehir='Istanbul'  
UNION  
SELECT isim ASisci_veya_sehir_ismi , maas  
FROM personel  
WHERE isim = 'Mehmet Ozturk';
```

ISCI_VEYA_SEHIR_ISMI	MAAS
Istanbul	4500
Istanbul	5500
Mehmet Ozturk	3500
Mehmet Ozturk	6000
Mehmet Ozturk	7000

NOT : 2.sorgunun sonuna ORDER BY komutunu kullanırsanız tüm tabloyu istediğiniz sıralamaya göre sıralar

```
ORDER BY maas;
```

ISCI_VEYA_SEHIR_ISMI	MAAS
Mehmet Ozturk	3500
Istanbul	4500
Istanbul	5500
Mehmet Ozturk	6000
Mehmet Ozturk	7000

UNION OPERATOR

3) Sehirlerden odenen ucret 3000'den fazla olanlari ve personelden ucreti 5000'den az olanlari bir tabloda maas miktarina gore sirali olarak gosteren sorguyu yaziniz

```
SELECT sehir ASisci_veya_sehir_ismi , maas  
FROM personel  
WHERE maas>3000  
UNION  
SELECT isim ASisci_veya_sehir_ismi , maas  
FROM personel  
WHERE maas<5000;
```

ISCI_VEYA_SEHIR_ISMI	MAAS
Ankara	3500
Ankara	4500
Ankara	7000
Bursa	4500
Hatice Sahin	4500
Istanbul	4500
Istanbul	5500
Izmir	6000
Mehmet Ozturk	3500
Veli Sahin	4500
Veli Sahin	4500

TECHFOODED

UNION OPERATOR

2 Tablodan Data Birlestirme

Personel isminde bir tablo olusturun.Icinde id,isim,sehir,maas ve sirket field'lari olsun. Id'yi 2.yontemle PK yapin

```
CREATE TABLE personel
(
    id number(9),
    isim varchar2(50),
    sehir varchar2(50),
    maas number(20),
    sirket varchar2(20),
    CONSTRAINT personel_pk PRIMARY KEY (id)
);
```

```
INSERT INTO personel VALUES(123456789, 'Ali Yilmaz', 'Istanbul', 5500, 'Honda');
INSERT INTO personel VALUES(234567890, 'Veli Sahin', 'Istanbul', 4500, 'Toyota');
INSERT INTO personel VALUES(345678901, 'Mehmet Ozturk', 'Ankara', 3500, 'Honda');
INSERT INTO personel VALUES(456789012, 'Mehmet Ozturk', 'Izmir', 6000, 'Ford');
INSERT INTO personel VALUES(567890123, 'Mehmet Ozturk', 'Ankara', 7000, 'Tofas');
INSERT INTO personel VALUES(456715012, 'Veli Sahin', 'Ankara', 4500, 'Ford');
INSERT INTO personel VALUES(123456710, 'Hatice Sahin', 'Bursa', 4500, 'Honda');
```

Personel_bilgi isminde bir tablo olusturun.Icinde id,tel ve cocuk sayisi field'lari olsun. Id'yi FK yapin ve personel tablosu ile relation kurun

```
CREATE TABLE personel_bilgi
(
    id number(9),
    tel char(10) UNIQUE ,
    cocuk_sayisi number(2),
    CONSTRAINT personel_bilgi_fk FOREIGN KEY (id) REFERENCES personel(id)
);
```

```
INSERT INTO personel_bilgi VALUES(123456789, '5302345678' , 5);
INSERT INTO personel_bilgi VALUES(234567890, '5422345678', 4);
INSERT INTO personel_bilgi VALUES(345678901, '5354561245', 3);
INSERT INTO personel_bilgi VALUES(456789012, '5411452659', 3);
INSERT INTO personel_bilgi VALUES(567890123, '5551253698', 2);
INSERT INTO personel_bilgi VALUES(456789012, '5524578574', 2);
INSERT INTO personel_bilgi VALUES(123456710, '5537488585', 1);
```

UNION OPERATOR

id'si 12345678 olan personelin Personel tablosundan sehir ve maasini, personel_bilgi tablosundan da tel ve cocuk sayisini yazdirin

```
SELECT sehir AS Sehir_tel ,maas AS cocuk_sayisi_veya_maas  
FROM personel  
WHERE id='123456789'
```

UNION

```
SELECT tel,cocuk_sayisi  
FROM personel_bilgi  
WHERE id= '123456789';
```

ID	ISIM	SEHIR	MAAS	SIRKET
123456789	Ali Yilmaz	Istanbul	5500	Honda
234567890	Veli Sahin	Istanbul	4500	Toyota
345678901	Mehmet Ozturk	Ankara	3500	Honda
456789012	Mehmet Ozturk	Izmir	6000	Ford
567890123	Mehmet Ozturk	Ankara	7000	Tofas
456789152	Veli Sahin	Ankara	4500	Ford
123456710	Hatice Sahin	Bursa	4500	Honda

SEHIR_TEL	COCUK SAYISI VEYA TEL
5302345678	5
Istanbul	5500

ID	TEL	COCUK SAYISI
123456789	5302345678	5
234567890	5422345678	4
345678901	5354561245	3
456789012	5411452659	3
567890123	5551253698	2
456789012	5524578574	2
123456710	5537488585	1

NOT : Union islemi yaparken

- 1) Her 2 QUERY'den elde edeceginiz tablolarin sutun sayiları esit olmali
- 2) Alt alta gelecek sutunların data type'ları aynı olmalı

UNION ALL OPERATOR

- 1) Personel tablosundada maasi 5000'den az olan tum isimleri ve maaslari bulunuz

```
SELECT isim,maas  
FROM personel  
WHERE maas<5000;
```

ISIM	MAAS
Veli Sahin	4500
Mehmet Ozturk	3500
Veli Sahin	4500
Hatice Sahin	4500

ID	ISIM	SEHIR	MAAS	SIRKET
123456789	Ali Yilmaz	Istanbul	5500	Honda
234567890	Veli Sahin	Istanbul	4500	Toyota
345678901	Mehmet Ozturk	Ankara	3500	Honda
456789012	Mehmet Ozturk	Izmir	6000	Ford
567890123	Mehmet Ozturk	Ankara	7000	Tofas
456715012	Veli Sahin	Ankara	4500	Ford
123456710	Hatice Sahin	Bursa	4500	Honda

T E C H P R O E D

UNION ALL OPERATOR

2) Ayni sorguyu UNION ile iki kere yazarak calistirin

```
SELECT sehir,maas  
FROM personel  
WHERE maas<5000
```

UNION

```
SELECT sehir,maas  
FROM personel  
WHERE maas<5000;
```



SEHIR	MAAS
Ankara	3500
Ankara	4500
Bursa	4500
Istanbul	4500

3) Ayni sorguyu UNION ALL ile iki kere yazarak calistirin

```
SELECT sehir,maas  
FROM personel  
WHERE maas<5000;
```

UNION ALL

```
SELECT sehir,maas  
FROM personel  
WHERE maas<5000;
```

SEHIR	MAAS
Istanbul	4500
Ankara	3500
Ankara	4500
Bursa	4500
Istanbul	4500
Ankara	3500
Ankara	4500
Bursa	4500

T E C H P O E D

UNION ALL OPERATOR

UNION islemi 2 veya daha cok **SELECT** isleminin sonuc **KUMELERINI** birlestirmek icin kullanilir,
Ayni kayit birden fazla olursa, sadece bir tanesini alir.

UNION ALL ise tekrarli elemanlari, tekrar sayisinda yazar.

NOT : UNION ALL ile birlestirmelerde de

- 1) Her 2 QUERY'den elde edeceginiz tablolarin sutun sayiları esit olmali
- 2) Alt alta gelecek sutunların data type'ları aynı olmalı

T E C H P R O E D

UNION ALL OPERATOR

1) Tabloda personel maasi 4000'den cok olan tum sehirleri ve maaslari yazdirin

SEHIR	MAAS
Istanbul	5500
Istanbul	4500
Izmir	6000
Ankara	7000
Ankara	4500
Bursa	4500

```
SELECT sehir,maas  
FROM personel  
WHERE maas>4000;
```

2) Tabloda personel maasi 5000'den az olan tum isimleri ve maaslari yazdirin

ISIM	MAAS
Veli Sahin	4500
Mehmet Ozturk	3500
Veli Sahin	4500
Hatice Sahin	4500

```
SELECT isim,maas  
FROM personel  
WHERE maas<5000;
```

3) Iki soruyu UNION ve UNION ALL ile birlestirin

ID	ISIM	SEHIR	MAAS	SIRKET
123456789	Ali Yilmaz	Istanbul	5000	Honda
234567890	Veli Sahin	Istanbul	5000	Toyota
345678901	Mehmet Ozturk	Ankara	4500	Honda
456789012	Mehmet Ozturk	Izmir	6000	Ford
567890123	Mehmet Ozturk	Ankara	6000	Tofas
456715012	Veli Sahin	Ankara	4500	Ford
123456710	Hatice Sahin	Bursa	4500	Honda

SEHIR	MAAS
Ankara	4500
Ankara	7000
Bursa	4500
Hatice Sahin	4500
Istanbul	4500
Istanbul	5500
Izmir	6000
Mehmet Ozturk	3500
Veli Sahin	4500

SEHIR	MAAS
Istanbul	5500
Istanbul	4500
Izmir	6000
Ankara	7000
Ankara	4500
Bursa	4500
Veli Sahin	4500
Mehmet Ozturk	3500
Veli Sahin	4500
Hatice Sahin	4500

TECHPRD

INTERSECT OPERATOR

- 1) Personel tablosundan Istanbul veya Ankara'da calisanların id'lerini yazdır

ID
123456789
234567890
345678901
567890123
456715012

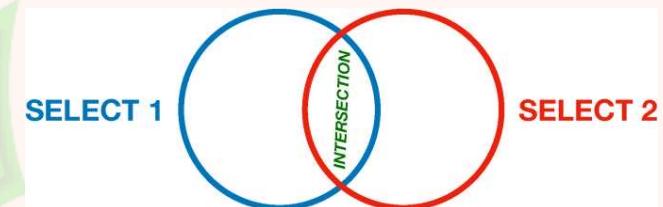
```
SELECT id  
FROM personel  
WHERE sehir IN ('Istanbul','Ankara');
```

ID
345678901
456789012
567890123
456789012

```
2)  
SELECT id  
FROM personel_bilgi  
WHERE cocuk_sayisi IN (2,3);
```

- 3) İki soruyu INTERSECT ile birleştirin

ID
345678901
567890123



T E C H I R O E D

INTERSECT OPERATOR

1) Maasi 4800'den az olanlar veya 5000'den cok olanların id'lerini listeleyin

ID
234567890
345678901
456789012
567890123
456715012
123456710

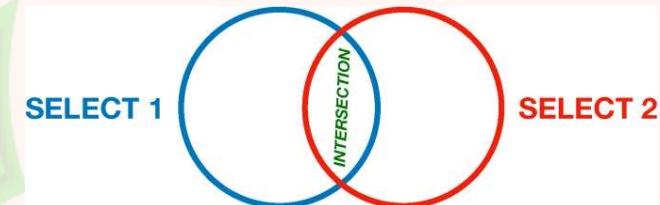
```
SELECT id  
FROM personel  
WHERE maas NOT BETWEEN 4800 AND 5500;
```

2) Personel_bilgi tablosundan 2 veya 3 cocugu olanların id lerini yazdirin

ID
345678901
456789012
567890123
456789012

```
SELECT id  
FROM personel_bilgi  
WHERE cocuk_sayisi IN (2,3);
```

3) Iki soruyu INTERSECT ile birlestirin



ID
345678901
456789012
567890123

T E C H F R O E D

INTERSECT OPERATOR

3) Honda,Ford ve Tofas'ta calisan ortak isimde personel varsa listeleyin

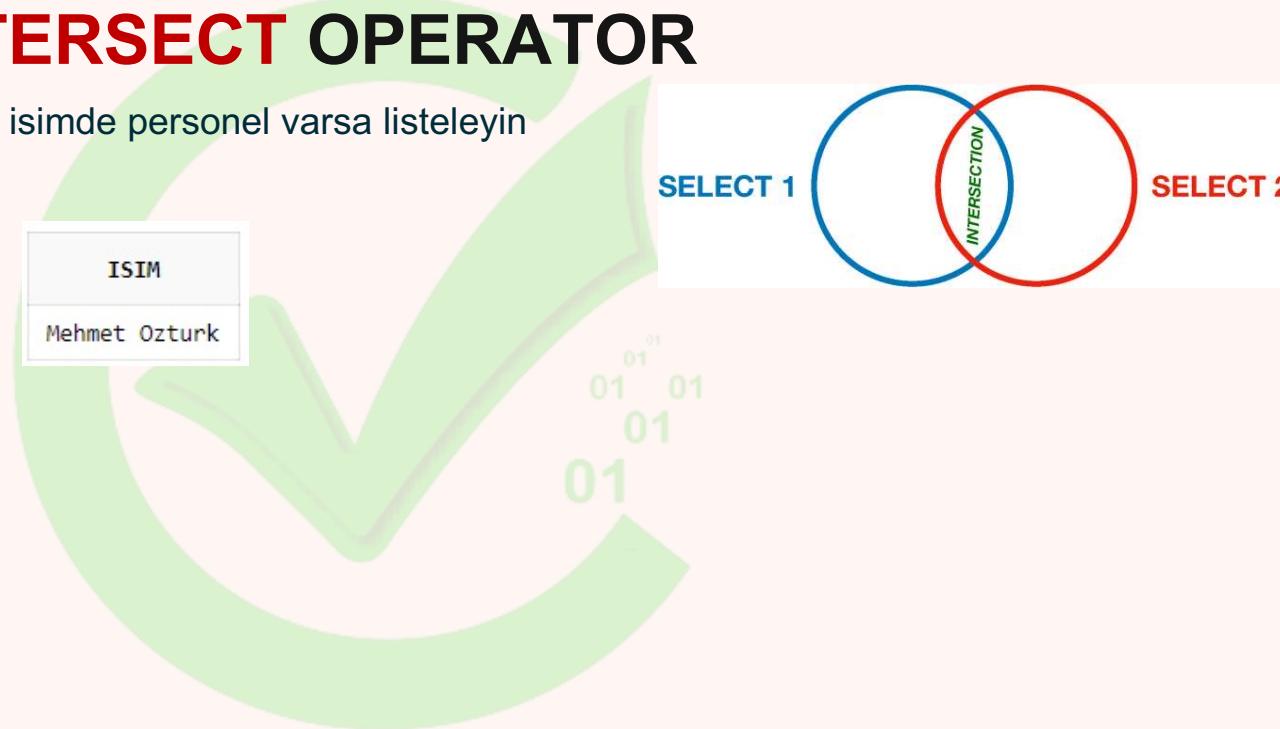
```
SELECT isim  
FROM personel  
WHERE sirket='Honda'
```

INTERSECT

```
SELECT isim  
FROM personel  
WHERE sirket='Ford'
```

INTERSECT

```
SELECT isim  
FROM personel  
WHERE sirket='Tofas';
```



T E C H P R O E D

MINUS OPERATOR

1) 5000'den az maas alip Honda'da calismayanları yazdirin

```
SELECT isim,sirket  
FROM personel  
WHERE maas<5000
```

MINUS

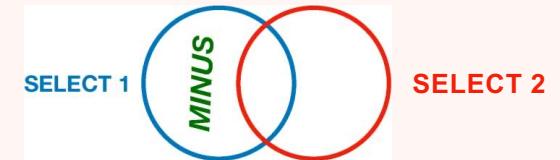
```
SELECT isim,sirket  
FROM personel  
WHERE sirket='Honda'
```

2) Ismi Mehmet Ozturk olup Istanbul'da calismayanların isimlerini ve sehirlerini listeleyin

```
SELECT isim,sehir  
FROM personel  
WHERE isim='Mehmet Ozturk'
```

MINUS

```
SELECT isim,sirket  
FROM personel  
WHERE sehir='Istanbul';
```



ISIM	SEHIR
Mehmet Ozturk	Ankara
Mehmet Ozturk	Izmir

T C H R O E D

JOINS

2 Tablodaki datalari Birlestirmek icin kullanilir.

Su ana kadar gordugumuz Union,Intersect ve Minus soru sonucları icin kullanilir
Tablolar icin ise **JOIN** kullanilir

5 Cesit Join vardir

- 1) INNER JOIN iki Tablodaki ortak datalari gosterir
- 2) LEFT JOIN Ilk datada olan tum recordlari gosterir
- 3) RIGHT JOIN Ikinci tabloda olan tum recordlari gosterir
- 4) FULL JOIN Iki tablodaki tum recordlari gosterir
- 5) SELF JOIN Bir tablonun kendi icinde Join edilmesi ile olusur.

T E C H P R O E D

INNER JOINS

```
CREATE TABLE sirketler
(
    sirket_id number(9),
    sirket_isim varchar2(20)
);
```

```
CREATE TABLE siparisler
(
    siparis_id number(9),
    sirket_id number(9),
    siparis_tarihi date
);
```



```
INSERT INTO sirketler VALUES(100, 'Toyota');
INSERT INTO sirketler VALUES(101, 'Honda');
INSERT INTO sirketler VALUES(102, 'Ford');
INSERT INTO sirketler VALUES(103, 'Hyundai');
```

SIRKET_ID	SIRKET_ISIM
100	Toyota
101	Honda
102	Ford
103	Hyundai

```
INSERT INTO siparisler VALUES(11, 101, '17-Apr-2020');
INSERT INTO siparisler VALUES(22, 102, '18-Apr-2020');
INSERT INTO siparisler VALUES(33, 103, '19-Apr-2020');
INSERT INTO siparisler VALUES(44, 104, '20-Apr-2020');
INSERT INTO siparisler VALUES(55, 105, '21-Apr-2020');
```

SIPARIS_ID	SIRKET_ID	SIPARIS_TARIHI
11	101	17-APR-20
22	102	18-APR-20
33	103	19-APR-20
44	104	20-APR-20
55	105	21-APR-20

INNER JOINS

SORU) İki Tabloda sirket_id'si aynı olanların sirket_ismi, siparis_id ve siparis_tarihleri ile yeni bir tablo oluşturun

```
SELECT sirketler.sirket_isim, siparisler. siparis_id, siparisler. siparis_tarihi  
FROM sirketler INNER JOIN siparisler  
ON sirketler.sirket_id = siparisler.sirket_id;
```

TABLE 1

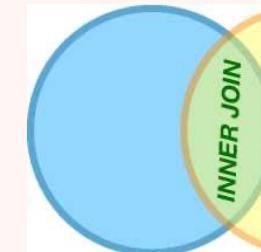


TABLE 2

SIRKET_ISIM	SIPARIS_ID	SIPARIS_TARIHI
Honda	11	17-APR-20
Ford	22	18-APR-20
Hyundai	33	19-APR-20

NOT :

- 1) Select'ten sonra tabloda gormek istediginiz sutunları yazarken **Tablo_adi.field_adi** şeklinde yazın
- 2) From'dan sonra tablo ismi yazarken **1.Tablo ismi + INNER JOIN + 2.Tablo ismi** yazmaliyiz
- 3) Join'i hangi kurala göre yapacaginiizi belirtmelisiniz. Bunun icin **ON+ kuralimiz** yazilmali

LEFT JOINS

TABLE 1

TABLE 2

```
SELECT sirketler.sirket_isim, siparisler. siparis_id, siparisler. siparis_tarihi  
FROM sirketler LEFT JOIN siparisler  
ON sirketler.sirket_id = siparisler.sirket_id;
```

SIRKET_ISIM	SIPARIS_ID	SIPARIS_TARIHI
Honda	11	17-APR-20
Ford	22	18-APR-20
Hyundai	33	19-APR-20
Toyota	-	-

NOT :

- 1) Left Join'de ilk tablodaki tum record'lar gosterilir.
- 2) Ilk tablodaki datalara 2.tablodan gelen ek datalar varsa bu ek datalar ortak datalar icin gosterilir ancak ortak olmayan datalar icin o kisimlar bos kalir
- 3) Ilk yazdiginiz Tablonun tamamini aldiği için hangi tabloyu istedigimize karar verip once onu yazmaliyiz

RIGHT JOINS

TABLE 1

TABLE 2

```
SELECT sirketler.sirket_isim, siparisler. siparis_id, siparisler. siparis_tarihi  
FROM sirketler RIGHT JOIN siparisler  
ON sirketler.sirket_id = siparisler.sirket_id;
```

SIRKET_ISIM	SIPARIS_ID	SIPARIS_TARIHI
Honda	11	17-APR-20
Ford	22	18-APR-20
Hyundai	33	19-APR-20
-	55	21-APR-20
-	44	20-APR-20

NOT :

- 1) Right Join'de ikinci tablodaki tum record'lar gösterilir.
- 2) Ikinci tablodaki datalara 1.tablodan gelen ek datalar varsa bu ek datalar ortak datalar icin gösterilir ancak ortak olmayan datalar icin o kisimlar bos kalir

FULL JOINS

```
SELECT sirketler.sirket_isim, siparisler. siparis_id, siparisler. siparis_tarihi  
FROM sirketler FULL JOIN siparisler  
ON sirketler.sirket_id = siparisler.sirket_id;
```

NOT :

- 1) FULL Join'de iki tabloda var olan tum record'lar gösterilir.
- 2) Bir tabloda olup otekinde olmayan data'lar bos kalır

SIRKET_ISIM	SIPARIS_ID	SIPARIS_TARIHI
Honda	11	17-APR-20
Ford	22	18-APR-20
Hyundai	33	19-APR-20
-	44	20-APR-20
-	55	21-APR-20
Toyota	-	-

T E C H P R O E D

SELF JOINS

```
CREATE TABLE personel
```

```
(  
    id number(2),  
    isim varchar2(20),  
    title varchar2(60),  
    yonetici_id number(2)  
)
```

```
INSERT INTO personel VALUES(1, 'Ali Can', 'SDET', 2);  
INSERT INTO personel VALUES(2, 'Veli Cem', 'QA', 3);  
INSERT INTO personel VALUES(3, 'Ayse Gul', 'QA Lead', 4);  
INSERT INTO personel VALUES(4, 'Fatma Can', 'CEO', 5);
```

ID	ISIM	TITLE	YONETICI_ID
1	Ali Can	SDET	2
2	Veli Cem	QA	3
3	Ayse Gul	QA Lead	4
4	Fatma Can	CEO	5

Her personelin yanina yonetici ismini yazdiran bir tablo olusturun

```
SELECT p1.isim AS personel_ismi, p2.isim AS yonetici_ismi  
FROM personel p1 INNER JOIN personel p2  
ON p1.yonetici_id = p2.id;
```

PERSONEL_ISMI	YONETICI_ISMI
Ali Can	Veli Cem
Veli Cem	Ayse Gul
Ayse Gul	Fatma Can

TECHPRUED

LIKE Condition

LIKE condition WHERE ile kullanilarak SELECT, INSERT, UPDATE, veya DELETE statement ile calisan wildcards'a izin verir.. Ve bize pattern matching yapma imkani verir.

```
CREATE TABLE musteriler
(
id number(10) UNIQUE,
isim varchar2(50) NOT NULL,
gelir number(6)
);
```

```
INSERT INTO musteriler (id, isim, gelir) VALUES (1001, 'Ali', 62000);
INSERT INTO musteriler (id, isim, gelir) VALUES (1002, 'Ayse', 57500);
INSERT INTO musteriler (id, isim, gelir) VALUES (1003, 'Feride', 71000);
INSERT INTO musteriler (id, isim, gelir) VALUES (1004, 'Fatma', 42000);
INSERT INTO musteriler (id, isim, gelir) VALUES (1005, 'Kasim', 44000);
```

1) % => 0 veya birden fazla karakter belirtir

SORU : Ismi A harfi ile baslayan musterilerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *
FROM musteriler
WHERE isim LIKE 'A%';
```

ID	ISIM	GELIR
1001	Ali	62000
1002	Ayse	57500

ID	ISIM	GELIR
1003	Feride	71000
1004	Fatma	42000
1005	Kasim	44000

LIKE Condition

SORU : Ismi e harfi ile biten musterilerin isimlerini ve gelir'lerini yazdiran QUERY yazin

```
SELECT isim,gelir  
FROM musteriler  
WHERE isim LIKE '%e';
```

ISIM	GELIR
Ayse	57500
Feride	71000

SORU : Isminin icinde er olan musterilerin isimlerini ve gelir'lerini yazdiran QUERY yazin

```
SELECT isim,gelir  
FROM musteriler  
WHERE isim LIKE '%er%';
```

ISIM	GELIR
Feride	71000

LIKE Condition

2) _ => sadece bir karakteri gosterir.

SORU : Ismi 5 harfli olup son 4 harfi atma olan musterilerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *
FROM musteriler
WHERE isim LIKE '_atma';
```

ID	ISIM	GELIR
1004	Fatma	42000

SORU : Ikinci harfi a olan musterilerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *
FROM musteriler
WHERE isim LIKE '_a%';
```

ID	ISIM	GELIR
1004	Fatma	42000
1005	Kasim	44000

SORU : Ucuncu harfi s olan musterilerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *
FROM musteriler
WHERE isim LIKE '__s%';
```

ID	ISIM	GELIR
1002	Ayse	57500
1005	Kasim	44000

LIKE Condition

SORU : Ucuncu harfi s olan ismi 4 harfli musterilerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *
FROM musteriler
WHERE isim LIKE '__s__';
```

ID	ISIM	GELIR
1002	Ayse	57500

SORU : Ilk harfi F olan en az 4 harfli musterilerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *
FROM musteriler
WHERE isim LIKE 'F_%_%%';
```

ID	ISIM	GELIR
1003	Feride	71000
1004	Fatma	42000

SORU : Ikinci harfi a,4.harfi m olan musterilerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *
FROM musteriler
WHERE isim LIKE '_a_m%';
```

ID	ISIM	GELIR
1004	Fatma	42000

LIKE Condition

3) [] REGEXP_LIKE => sadece bir karakteri gosterir.

```
CREATE TABLE kelimeler  
(  
    id number(10) UNIQUE,  
    kelime varchar2(50) NOT NULL,  
    Harf_sayisi number(6)  
);
```

```
INSERT INTO kelimeler VALUES (1001, 'hot', 3);  
INSERT INTO kelimeler VALUES (1002, 'hat', 3);  
INSERT INTO kelimeler VALUES (1003, 'hit', 3);  
INSERT INTO kelimeler VALUES (1004, 'hbt', 3);  
INSERT INTO kelimeler VALUES (1008, 'hct', 3);  
INSERT INTO kelimeler VALUES (1005, 'adem', 4);  
INSERT INTO kelimeler VALUES (1006, 'selim', 5);  
INSERT INTO kelimeler VALUES (1007, 'yusuf', 5);
```

SORU : Ilk harfi h,son harfi t olup 2.harfi a veya i olan 3 harflili kelimelerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM kelimeler  
WHERE REGEXP_LIKE (kelime, 'h[ai]t');
```

LIKE Condition

SORU : İlk harfi h, son harfi t olup 2.harfi a ile k arasında olan 3 harfli kelimelerin tüm bilgilerini yazdırın QUERY yazın

```
SELECT *  
FROM kelimeler  
WHERE REGEXP_LIKE (kelime, 'h[a-k]t');
```

ID	KELIME	HARF_SAYISI
1002	hat	3
1003	hit	3
1004	hbt	3
1008	hct	3

SORU : İçinde m veya i olan kelimelerin tüm bilgilerini yazdırın QUERY yazın

```
SELECT *  
FROM kelimeler  
WHERE REGEXP_LIKE (kelime, '[mi](*)'); [a|n] de olur
```

ID	KELIME	HARF_SAYISI
1003	hit	3
1005	adem	4
1006	selim	5

SORU : a veya s ile başlayan kelimelerin tüm bilgilerini yazdırın QUERY yazın

```
SELECT *  
FROM kelimeler  
WHERE REGEXP_LIKE (kelime, '^[as] ');
```

ID	KELIME	HARF_SAYISI
1005	adem	4
1006	selim	5

LIKE Condition

SORU : m veya f ile biten kelimelerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM kelimeler  
WHERE REGEXP_LIKE (kelime, '[ea]$');
```

ID	KELIME	HARF SAYISI
1005	adem	4
1006	selim	5
1007	yusuf	5

T E C H P R O E D

NOT LIKE Condition

SORU 1 : ilk harfi h olmayan kelimelerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM kelimeler  
WHERE kelime NOT LIKE 'h%';
```

ID	KELIME	HARF_SAYISI
1005	adem	4
1006	selim	5
1007	yusuf	5

01

SORU 2 : a harfi icermeyen kelimelerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM kelimeler  
WHERE kelime NOT LIKE '%a%';
```

ID	KELIME	HARF_SAYISI
1001	hot	3
1003	hit	3
1004	hbt	3
1008	hct	3
1006	selim	5
1007	yusuf	5

TECH

CODE

NOT LIKE Condition

SORU 3 : ikinci ve ucuncu harfi 'de' olmayan kelimelerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM kelimeler  
WHERE kelime NOT LIKE '_de%';
```

ID	KELIME	HARF_SAYISI
1001	hot	3
1002	hat	3
1003	hit	3
1004	hbt	3
1008	hct	3
1006	selim	5
1007	yusuf	5

SORU 4 : 2. harfi e,i veya o olmayan kelimelerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM kelimeler  
WHERE NOT REGEXP_LIKE (kelime, '[_eio]');
```

ID	KELIME	HARF_SAYISI
1002	hat	3
1004	hbt	3
1008	hct	3
1007	yusuf	5

UPPER – LOWER - INITCAP

Tablolari yazdirirken buyuk harf, kucuk harf veya ilk harfleri buyuk digerleri kucuk harf yazdirmak icin kullaniriz

`SELECT UPPER(kelime)
FROM kelimeler;`

UPPER(KELIME)
HOT
HAT
HIT
HBT
HCT
ADEM
SELIM
YUSUF

`SELECT LOWER(kelime)
FROM kelimeler;`

LOWER(KELIME)
hot
hat
hit
hbt
hct
adem
selim
yusuf

`SELECT INITCAP(kelime)
FROM kelimeler;`

INITCAP(KELIME)
Hot
Hat
Hit
Hbt
Hct
Adem
Selim
Yusuf

TECHPRO

DISTINCT

URUN_ID	MUSTERI_ISIM	URUN_ISIM
10	Ali	Portakal
10	Ali	Portakal
20	Veli	Elma
30	Ayse	Armut
20	Ali	Elma
10	Adem	Portakal
40	Veli	Kaysi
20	Elif	Elma

SELECT DISTINCT urun_isim
FROM musteri_urun;

URUN_ISIM
Elma
Portakal
Kaysi
Armut

SELECT DISTINCT musteri_isim
FROM musteri_urun;

MUSTERI_ISIM
Veli
Ayse
Elif
Adem
Ali

Tabloda kaç farklı meyve vardır ?

SELECT COUNT(DISTINCT urun_isim) AS urun_cesit_sayisi
FROM musteri_urun;

URUN_CESIT_SAYISI
4

T E C H P R O J E C T

FETCH NEXT (SAYI) ROW ONLY- OFFSET

1) Tabloyu urun_id ye gore siralayiniz

URUN_ID	MUSTERI_ISIM	URUN_ISIM
10	Ali	Portakal
10	Ali	Portakal
10	Adem	Portakal
20	Veli	Elma
20	Elif	Elma
20	Ali	Elma
30	Ayse	Armut
40	Veli	Kaysi

2) Sirali tablodan ilk 3 kaydi listeleyin

```
SELECT *  
FROM musteri_urun  
ORDER BY urun_id  
FETCH NEXT 3 ROW ONLY;
```

URUN_ID	MUSTERI_ISIM	URUN_ISIM
10	Ali	Portakal
10	Adem	Portakal
10	Ali	Portakal

3) Sirali tablodan 4. kayittan 7.kayida kadar olan kayitlari listeleyin

```
SELECT *  
FROM musteri_urun  
ORDER BY urun_id  
OFFSET 3 ROW  
FETCH NEXT 4 ROW ONLY;
```

URUN_ID	MUSTERI_ISIM	URUN_ISIM
20	Veli	Elma
20	Elif	Elma
20	Ali	Elma
30	Ayse	Armut

TECHPROED

PIVOT CLAUSES

```
CREATE TABLE musteri_urun  
(  
    urun_id number(10),  
    musteri_isim varchar2(50),  
    urun_isim varchar2(50)  
);
```

```
SELECT *  
FROM (SELECT urun_isim, musteri_isim FROM musteri_urun)  
PIVOT (COUNT(urun_isim) FOR urun_isim IN ('Portakal', 'Elma', 'Kaysi', 'Armut'));
```

```
INSERT INTO musteri_urun VALUES (10, 'Ali', 'Portakal');  
INSERT INTO musteri_urun VALUES (10, 'Ali', 'Portakal');  
INSERT INTO musteri_urun VALUES (20, 'Veli', 'Elma');  
INSERT INTO musteri_urun VALUES (30, 'Ayse', 'Armut');  
INSERT INTO musteri_urun VALUES (20, 'Ali', 'Elma');  
INSERT INTO musteri_urun VALUES (10, 'Adem', 'Portakal');  
INSERT INTO musteri_urun VALUES (40, 'Veli', 'Kaysi');  
INSERT INTO musteri_urun VALUES (20, 'Elif', 'Elma');
```

MUSTERI_ISIM	'Portakal'	'Elma'	'Kaysi'	'Armut'
Veli	0	1	1	0
Ayse	0	0	0	1
Elif	0	1	0	0
Adem	1	0	0	0
Ali	2	1	0	0

```
SELECT *  
FROM (SELECT urun_isim, musteri_isim FROM musteri_urun)  
PIVOT (COUNT(musteri_isim) FOR musteri_isim IN ('Ali', 'Veli', 'Ayse', 'Adem', 'Elif'));
```

URUN_ISIM	'Ali'	'Veli'	'Ayse'	'Adem'	'Elif'
Elma	1	1	0	0	1
Portakal	2	0	0	1	0
Kaysi	0	1	0	0	0
Armut	0	0	1	0	0

ALTER TABLE STATEMENT

ALTER TABLE statement tabloda add, modify, veya drop/delete columns islemleri icin kullanilir.

ALTER TABLE statement tablolari yeniden isimlendirmek icin de kullanilir.

```
CREATE TABLE personel
```

```
(  
    id number(9),  
    isim varchar2(50),  
    sehir varchar2(50),  
    maas number(20),  
    sirket varchar2(20),  
    CONSTRAINT personel_pk PRIMARY KEY (id)  
);
```

```
INSERT INTO personel VALUES(123456789, 'Ali Yilmaz', 'Istanbul', 5500, 'Honda');  
INSERT INTO personel VALUES(234567890, 'Veli Sahin', 'Istanbul', 4500, 'Toyota');  
INSERT INTO personel VALUES(345678901, 'Mehmet Ozturk', 'Ankara', 3500, 'Honda');  
INSERT INTO personel VALUES(456789012, 'Mehmet Ozturk', 'Izmir', 6000, 'Ford');  
INSERT INTO personel VALUES(567890123, 'Mehmet Ozturk', 'Ankara', 7000, 'Tofas');  
INSERT INTO personel VALUES(456715012, 'Veli Sahin', 'Ankara', 4500, 'Ford');  
INSERT INTO personel VALUES(123456710, 'Hatice Sahin', 'Bursa', 4500, 'Honda');
```

ID	ISIM	SEHIR	MAAS	SIRKET
123456789	Ali Yilmaz	Istanbul	5500	Honda
234567890	Veli Sahin	Istanbul	4500	Toyota
345678901	Mehmet Ozturk	Ankara	3500	Honda
456789012	Mehmet Ozturk	Izmir	6000	Ford
567890123	Mehmet Ozturk	Ankara	7000	Tofas
456715012	Veli Sahin	Ankara	4500	Ford
123456710	Hatice Sahin	Bursa	4500	Honda

ALTER TABLE STATEMENT

1) ADD default değer ile tabloya bir sutun ekleme

ALTER TABLE personel

ADD ulke_isim varchar2(20) DEFAULT 'Turkiye';

ID	ISIM	SEHIR	MAAS	SIRKET	ULKE_ISIM
123456789	Ali Yilmaz	Istanbul	5500	Honda	Turkiye
234567890	Veli Sahin	Istanbul	4500	Toyota	Turkiye
345678901	Mehmet Ozturk	Ankara	3500	Honda	Turkiye
456789012	Mehmet Ozturk	Izmir	6000	Ford	Turkiye
567890123	Mehmet Ozturk	Ankara	7000	Tofas	Turkiye
456715012	Veli Sahin	Ankara	4500	Ford	Turkiye
123456710	Hatice Sahin	Bursa	4500	Honda	Turkiye

2) Tabloya birden fazla sutun ekleme

ALTER TABLE personel

ADD (cinsiyet varchar2(20) , yas number(3));

ID	ISIM	SEHIR	MAAS	SIRKET	ULKE_ISIM	CINSIYET	YAS
123456789	Ali Yilmaz	Istanbul	5500	Honda	Turkiye	-	-
234567890	Veli Sahin	Istanbul	4500	Toyota	Turkiye	-	-
345678901	Mehmet Ozturk	Ankara	3500	Honda	Turkiye	-	-
456789012	Mehmet Ozturk	Izmir	6000	Ford	Turkiye	-	-
567890123	Mehmet Ozturk	Ankara	7000	Tofas	Turkiye	-	-
456715012	Veli Sahin	Ankara	4500	Ford	Turkiye	-	-
123456710	Hatice Sahin	Bursa	4500	Honda	Turkiye	-	-

T E C H P

ALTER TABLE STATEMENT

3) DROP tablodan sutun silme

ALTER TABLE personel
DROP COLUMN yas;

ID	ISIM	SEHIR	MAAS	SIRKET	ULKE_ISIM	CINSIYET
123456789	Ali Yilmaz	Istanbul	5500	Honda	Turkiye	-
234567890	Veli Sahin	Istanbul	4500	Toyota	Turkiye	-
345678901	Mehmet Ozturk	Ankara	3500	Honda	Turkiye	-
456789012	Mehmet Ozturk	Izmir	6000	Ford	Turkiye	-
567890123	Mehmet Ozturk	Ankara	7000	Tofas	Turkiye	-
456715012	Veli Sahin	Ankara	4500	Ford	Turkiye	-
123456710	Hatice Sahin	Bursa	4500	Honda	Turkiye	-

4) RENAME COLUMN sutun adi degistirme

ALTER TABLE personel
RENAME COLUMN ulke_isim TO ulke_adi;

ID	ISIM	SEHIR	MAAS	SIRKET	ULKE_ADI	CINSIYET
123456789	Ali Yilmaz	Istanbul	5500	Honda	Turkiye	-
234567890	Veli Sahin	Istanbul	4500	Toyota	Turkiye	-
345678901	Mehmet Ozturk	Ankara	3500	Honda	Turkiye	-
456789012	Mehmet Ozturk	Izmir	6000	Ford	Turkiye	-
567890123	Mehmet Ozturk	Ankara	7000	Tofas	Turkiye	-
456715012	Veli Sahin	Ankara	4500	Ford	Turkiye	-
123456710	Hatice Sahin	Bursa	4500	Honda	Turkiye	-

TECH

DATA

ALTER TABLE STATEMENT

5) RENAME tablonun ismini degistirme

ALTER TABLE personel
RENAME TO isciler;

6) MODIFY sutunlarin ozelliklerini degistirme

ALTER TABLE isciler
MODIFY ulke_adi varchar2(30) NOT NULL;



01

Columns			
#	Column	Type	Length
1	ID	NUMBER	22
2	ISIM	VARCHAR2	50
3	SEHIR	VARCHAR2	50
4	MAAS	NUMBER	22
5	SIRKET	VARCHAR2	20
6	ULKE_ADI	VARCHAR2	30
7	CINSIYET	VARCHAR2	20

T E

D

INTERVIEW QUESTION

```
CREATE TABLE personel
```

```
(  
    id number(9),  
    isim varchar2(50),  
    sehir varchar2(50),  
    maas number(20),  
    sirket varchar2(20)  
);
```

```
INSERT INTO personel VALUES(123456789, 'Johnny Walk', 'New Hampshire', 2500, 'IBM');  
INSERT INTO personel VALUES(234567891, 'Brian Pitt', 'Florida', 1500, 'LINUX');  
INSERT INTO personel VALUES(245678901, 'Eddie Murphy', 'Texas', 3000, 'WELLS FARGO');  
INSERT INTO personel VALUES(456789012, 'Teddy Murphy', 'Virginia', 1000, 'GOOGLE');  
INSERT INTO personel VALUES(567890124, 'Eddie Murphy', 'Massachuset', 7000, 'MICROSOFT');  
INSERT INTO personel VALUES(456789012, 'Brad Pitt', 'Texas', 1500, 'TD BANK');  
INSERT INTO personel VALUES(123456719, 'Adem Stone', 'New Jersey', 2500, 'IBM');
```

```
CREATE TABLE isciler
```

```
(  
    id number(9),  
    isim varchar2(50),  
    sehir varchar2(50),  
    maas number(20),  
    sirket varchar2(20)  
);
```

```
INSERT INTO isciler VALUES(123456789, 'John Walker', 'Florida', 2500, 'IBM');  
INSERT INTO isciler VALUES(234567890, 'Brad Pitt', 'Florida', 1500, 'APPLE');  
INSERT INTO isciler VALUES(345678901, 'Eddie Murphy', 'Texas', 3000, 'IBM');  
INSERT INTO isciler VALUES(456789012, 'Eddie Murphy', 'Virginia', 1000, 'GOOGLE');  
INSERT INTO isciler VALUES(567890123, 'Eddie Murphy', 'Texas', 7000, 'MICROSOFT');  
INSERT INTO isciler VALUES(456789012, 'Brad Pitt', 'Texas', 1500, 'GOOGLE');  
INSERT INTO isciler VALUES(123456710, 'Mark Stone', 'Pennsylvania', 2500, 'IBM');
```

INTERVIEW QUESTION

1) Her iki tablodaki ortak id'leri ve personel tablosunda bu id'ye sahip isimleri listeleyen query yaziniz

```
SELECT isim,id  
FROM personel  
WHERE id IN (SELECT id  
              FROM isciler  
              WHERE isciler.id=personel.id);
```

ISIM	ID
Johnny Walk	123456789
Teddy Murphy	456789012
Brad Pitt	456789012

2) Her iki tablodaki ortak id ve isme sahip kayitlari listeleyen query yaziniz

```
SELECT isim,id  
FROM personel
```

INTERSECT

```
SELECT isim,id  
FROM personel;
```

ISIM	ID
Brad Pitt	456789012

INTERVIEW QUESTION

3) Personel tablosunda kac farkli sehirden personel var?

SELECT COUNT (DISTINCT sehir) AS sehir_sayisi
FROM personel;

SEHIR SAYISI
5

4) Personel tablosunda id'si cift sayi olan personel'in tum bilgilerini listeleyen Query yaziniz

SELECT *
FROM personel
WHERE MOD (id,2)=0;

ID	ISIM	SEHIR	MAAS	SIRKET
456789012	Teddy Murphy	Virginia	1000	GOOGLE
456789012	Brad Pitt	Texas	1500	TD BANK

TECHTRUE

INTERVIEW QUESTION

5) Personel tablosunda kac tane kayit oldugunu gosteren query yazin

```
SELECT COUNT(*)  
FROM personel;
```

COUNT(*)
6

```
SELECT COUNT(id) AS kayit_sayisi  
FROM personel;
```

KAYIT_SAYISI
6

6) Isciler tablosunda en yuksek maasi alan kisinin tum bilgilerini gosteren query yazin

Max Maas

```
SELECT MAX(maas) AS max_maas  
FROM isciler;
```

```
SELECT *  
FROM isciler  
WHERE maas IN (SELECT MAX(maas)  
FROM isciler);
```

ID	ISIM	SEHIR	MAAS	SIRKET
567890123	Eddie Murphy	Texas	7000	MICROSOFT

T E U H P R U E D

INTERVIEW QUESTION

7) Personel tablosunda en dusuk maasi alan kisinin tum bilgilerini gosteren query yazin

```
SELECT *  
FROM personel  
ORDER BY maas  
FETCH NEXT 1 ROW ONLY;
```

ID	ISIM	SEHIR	MAAS	SIRKET
456789012	Teddy Murphy	Virginia	1000	GOOGLE

8) Isciler tablosunda ikinci en yuksek maasi maasi gosteren query yazin

```
SELECT MAX(maas)  
FROM personel  
WHERE maas <> (SELECT MAX(maas)  
                  FROM personel);
```

MAX(MAAS)
2500

C H P R O E D

INTERVIEW QUESTION

9) Isciler tablosunda ikinci en dusuk maasi alan iscinin tum bilgilerini gosteren query yazin

```
SELECT *  
FROM isciler  
ORDER BY maas  
OFFSET 1 ROW  
FETCH NEXT 1 ROW ONLY;
```

ID	ISIM	SEHIR	MAAS	SIRKET
234567890	Brad Pitt	Florida	1500	APPLE

ID	ISIM	SEHIR	MAAS	SIRKET
123456789	John Walker	Florida	2500	IBM
234567890	Brad Pitt	Florida	1500	APPLE
345678901	Eddie Murphy	Texas	3000	IBM
456789012	Eddie Murphy	Virginia	1000	GOOGLE
567890123	Eddie Murphy	Texas	7000	MICROSOFT
456789012	Brad Pitt	Texas	1500	GOOGLE
123456710	Mark Stone	Pennsylvania	2500	IBM

ID	ISIM	SEHIR	MAAS	SIRKET
456789012	Eddie Murphy	Virginia	1000	GOOGLE
234567890	Brad Pitt	Florida	1500	APPLE
456789012	Brad Pitt	Texas	1500	GOOGLE
123456710	Mark Stone	Pennsylvania	2500	IBM
123456789	John Walker	Florida	2500	IBM
345678901	Eddie Murphy	Texas	3000	IBM
567890123	Eddie Murphy	Texas	7000	MICROSOFT

TECHP

DO

INTERVIEW QUESTION

10) Isciler tablosunda en yuksek maasi alan iscinin disindaki tum iscilerin, tum bilgilerini gosteren query yazin

```
SELECT *
FROM isciler
WHERE maas <> ( SELECT MAX(maas)
                  FROM isciler)
ORDER BY maas DESC;
```

ID	ISIM	SEHIR	MAAS	SIRKET
345678901	Eddie Murphy	Texas	3000	IBM
123456710	Mark Stone	Pennsylvania	2500	IBM
123456789	John Walker	Florida	2500	IBM
234567890	Brad Pitt	Florida	1500	APPLE
456789012	Brad Pitt	Texas	1500	GOOGLE
456789012	Eddie Murphy	Virginia	1000	GOOGLE

TECHPROED