SQL

Structured Query Language

(Yapılandırılmış Sorgu Dili)

1. Ders

Ayşe Sezen

Full-Stack Automation Engineer

BU DERS NELER ÖĞRENECEĞİZ?

1) Database Nedir ?

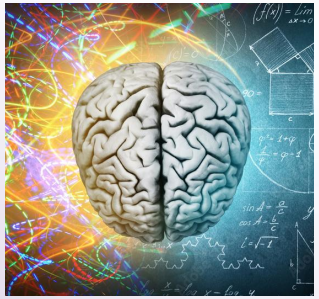
2) Genel Database Kavramları

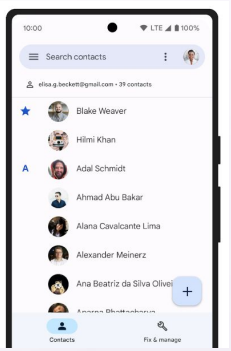
3) SQL'e giriş

Database (veritabanı) Nedir?


```java
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 surname");
        String email = scan.nextLine();

        System.out.println("Enter your pasword");
        String password = scan.nextLine();
```

Database (veritabanı) Nedir?





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

Veritabanı genellikle bir Veritabanı Yönetim Sistemi DBMS (Data Base Management System) ile kontrol edilir.

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

Database'in Faydaları Nelerdir?

1) Yüksek miktarda bilgi depolanabilir.

2) Oluşturma, Okuma, Değiştirme ve Silme kolaylığı

Create,Read,Update,Delete (CRUD)

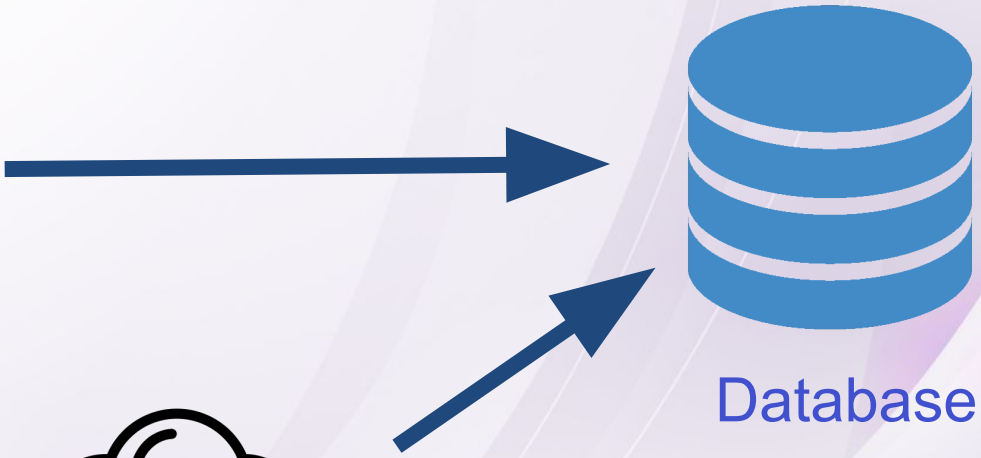
3) Girişin kolay ve kontrollü olması

4) Veriye ulaşım kolaylığı

5) Güvenlik

Database Doğrulama(Validation) Testi

User Interface (UI)



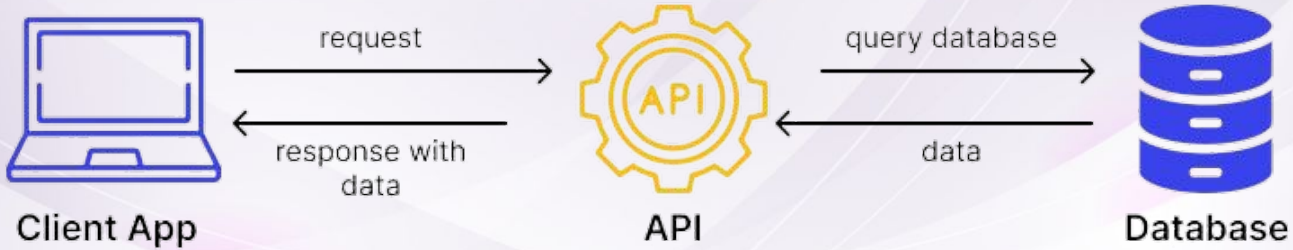
Database

Application Programming Interface (API)

API

Uygulama programlama arayüzü, bir yazılımın başka bir yazılımda tanımlanmış işlevlerini kullanabilmesi için oluşturulmuş bir arayüzdür.

API; web uygulaması, işletim sistemi, veritabanı, donanımlar yahut yazılım kütüphanesi için kullanılabilir



END To END (E2E) Testing

Wise Quarter

1) Eğer datayı User Interface (UI) kullanarak oluşturduysanız;
   a. Datayı UI'dan arama fonksiyonunu kullanarak doğrula (Selenium)
   b. Datayı SQL kodlarını kullanarak doğrula (SQL + Selenium)
   c. Datayı API kodlarını kullanarak doğrula (API + Selenium)

2) Eğer datayı SQL kodlarını kullanarak oluşturduysanız;
   a. Datayı UI'dan arama fonksiyonunu kullanarak doğrula (Selenium)
   b. Datayı SQL kodlarını kullanarak doğrula (SQL + Selenium)
   c. Datayı API kodlarını kullanarak doğrula (API + Selenium)

3) Eğer datayı API kodlarını kullanarak oluşturduysanız;
   a. Datayı UI'dan arama fonksiyonunu kullanarak doğrula (Selenium)
   b. Datayı SQL kodlarını kullanarak doğrula (SQL + Selenium)
   c. Datayı API kodlarını kullanarak doğrula (API + Selenium)

Database Management System (DBMS)

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

- Database'e erişimi düzenler.
- Create,Read,Update,Delete işlemlerini düzenler.
- Data güvenliğini sağlar.
- Formlar oluşturur ve formları işler.
- Sorgular oluşturur ve sorgular iletir.
- Raporlar oluşturur ve raporları işletir.
- Uygulamayı kontrol eder.
- Diğer uygulamalarla (Application) iletişimi sağlar.

Tablolar (Tables)

Wise Quarter

Headers →

Row(Record) →

Row(Record) →

Row(Record) →

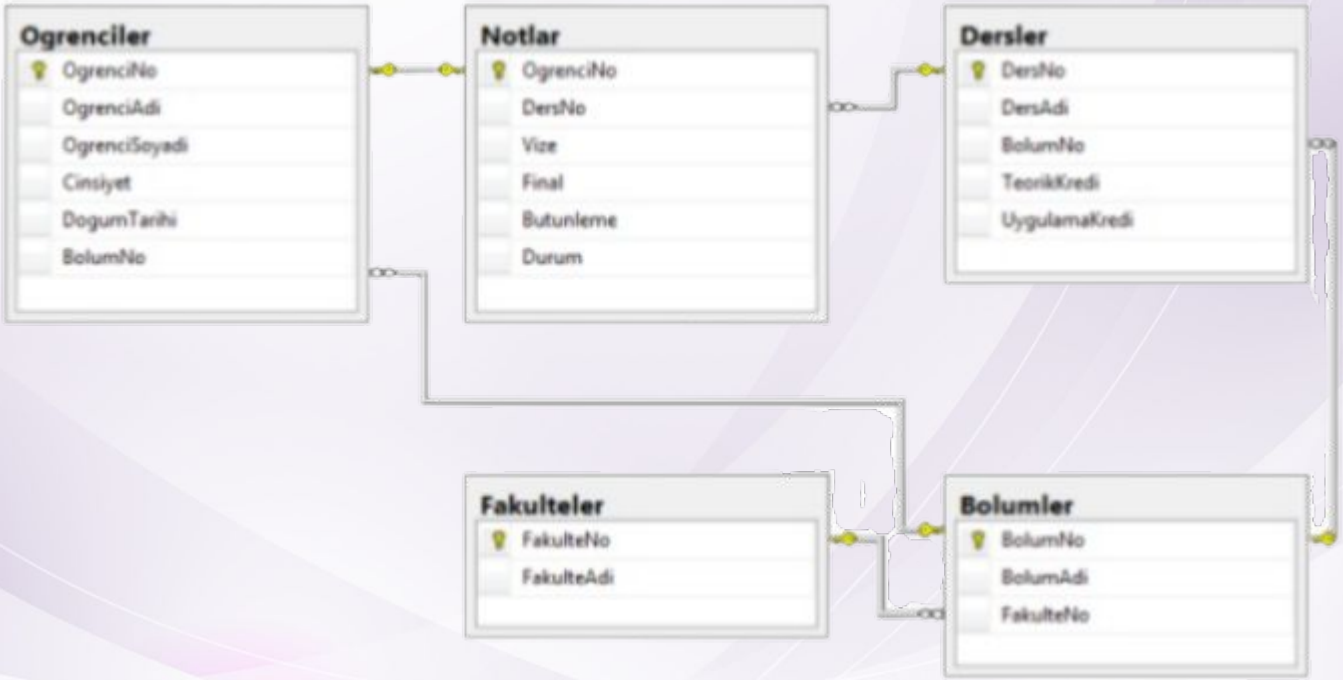
	A	B	C	D	E	F
3	**Emp Id**	**First Name**	**Last Name**	**Department**	**Location**	
4	101	Donald	Patrick	Finance	Banglore	
5	102	Samuel	Samson	Marketing	Hyderabad	
6	103	Ian	Jacob	Finance	Hyderabad	
7	104	David	Johnson	Marketing	Pune	
8	105	Ian	Smith	Marketing	Banglore	
9	106	Henry	Madrid	IT	Pune	
10	107	Ronica	Brave	Finance	Hyderabad	
11	108	Christine	Salvi	Marketing	Banglore	
12	109	Andrew	Baisley	IT	Hyderabad	
13	110	Erica	Irons	IT	Pune	
14						

Column (Field)

Column (Field)

Column (Field)

Relational Databases (İlişkili Tablolar)



Relational Databases (İlişkili Tablolar)

- **SQL tablolar** dataları ilişkili tablolarda depolar.
- Tablolar arası ilişkiler net olmalıdır.
- Tablolar arası geçiş kolay olmalıdır.
- Tabloların ve ilişkilerin bütününe **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_adi
1	Matematik
2	Tarih
3	Edebiyat
4	Yazılım
5	İstatistik

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

Çok Kullanılan Relational Databases

SQL Server : Microsoft tarafından geliştirilmiştir.

Negatif: Pahalı – Kurumsal Kullanıcılar için binlerce dolar ödenmesi gereklidir.
Pozitif : Zengin bir user interface'e sahip ve çok büyük dataların kullanılmasında sorunsuz çalışır.

MySQL Server : İsveçli MySQL firması tarafından geliştirildi. 2010'da Oracle satın aldı.

Negatif: Eş zamanlı çok fazla işlem girildiğinde çalışmayı durdurabilir.
Pozitif: Açık kaynak. Online destek ve ücretsiz çok fazla doküman var.

Çok Kullanılan Relational Databases

PostgreSQL Server : Bilgisayar bilimleri profesörü Michael Stonebraker tarafından tasarlandı.

Negatif: Kurulum ve ayarlar zor. Yeni başlayanlar için kullanımı zor.
Pozitif: Yeni nesil olarak ortaya çıktı. Kişiselleştirme mümkündür, zor görevler için ideal olabilir.

ORACLE DATABASE

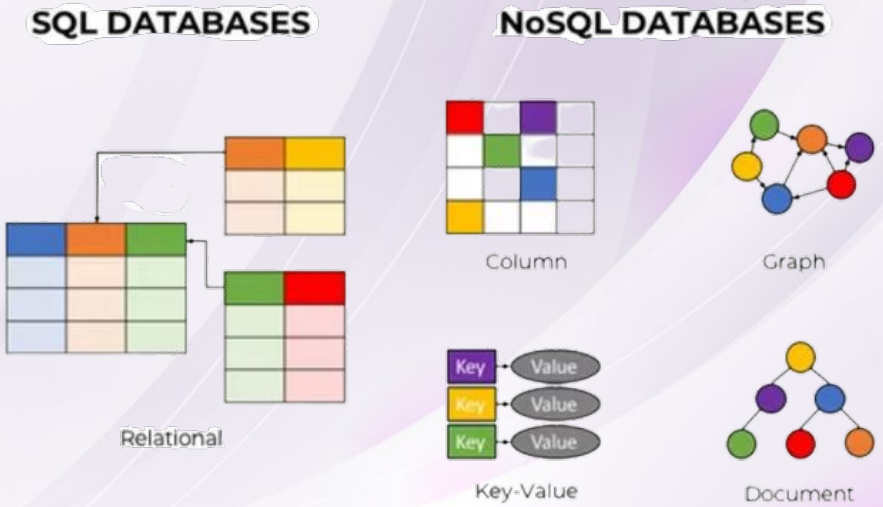
PL/SQL : Oracle database sunucuları içinde depolanır. PL/SQL SQL komutlarını özellikle karşılamak üzere dizayn edilmiştir.

Pozitif: PL/SQL yüksek güvenlik seviyesi sağlar ve Object-Oriented Programing'e (OOP) uyumludur.

Non-Relational Databases (NoSQL Databases)

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 Sorgulama Dili (Data Query Language - DQL)

DQL içindeki SELECT komutu ile veritabanında yer alan mevcut kayıtların bir kısmını veya tamamınını 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 Tanimlama 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 veritabanını veya veritabanı içindeki tabloyu günceller.

DROP : Bir veritabanını veya veritabanı içindeki tabloyu siler.

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

DCL komutları ile kullanıcılara 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.

SQL

Structured Query Language

(Yapılandırılmış Sorgu Dili)

2. Ders

Ayşe Sezen

Full-Stack Automation Engineer

BU DERS NELER ÖĞRENECEĞİZ?

1) Primary Key

2) Foreign Key

3) Data Tipleri

4) Tablo Oluşturma

Primary Key

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

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

Primary Keys

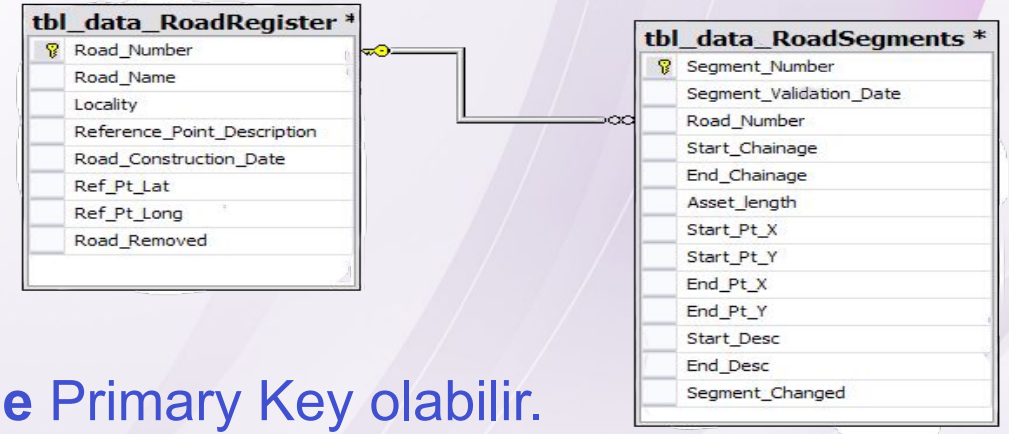
Employee ID	SURNAME	GIVEN NAME	MIDDLE NAME
8001000000	Smith	Jennifer	Abad
8001000001	Smith	John Nhiel	Galvez
8001000002	Dela Cruz	RJ	Prachaya
8001000003	Reyes	Gab	Ugalino
8001000004	Doe	RJ	Mendoza
8001000005	Licauco	David	Galvez

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

Primary Key

Primary Key değeri boş geçilemez ve **NULL** değer alamaz.

İlişkili veritabanlarında (relational database) mutlaka birincil anahtar olmalıdır.



- Bir Tabloda yalnızca **bir tane** Primary Key olabilir.
- Primary Key **benzersiz** (Unique) olmalıdır ama her Unique data Primary Key değildir.
- Primary Key her türlü datayı içerebilir. Sayı, String, ...
- Her tabloda Primary Key olması zorunlu değildir.

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.

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

SSN

Aadhar

EMP_ID	NAME	SSN
123	ABC	123-456-7980
124	XYZ	789-345-5678

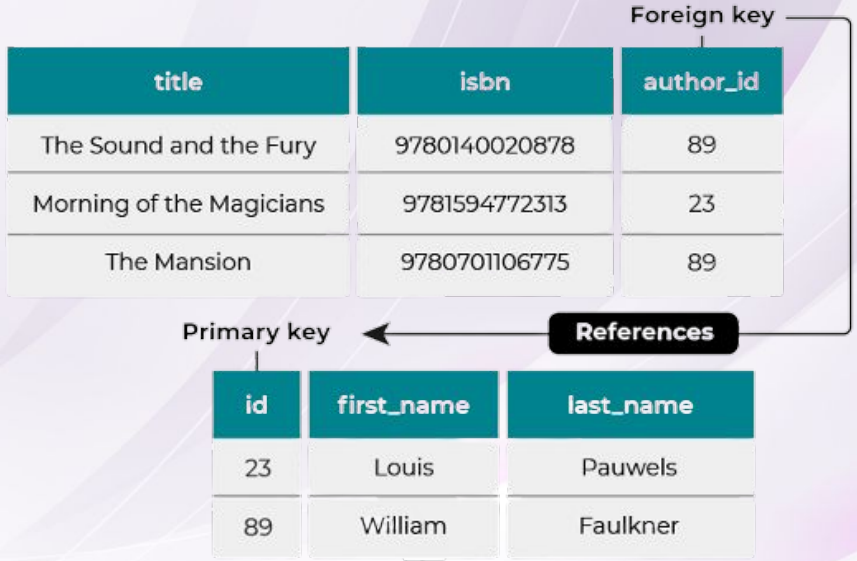
ID	EMP_ID	NAME	Department	START_DATE	END_DATE	Flag
1	123	ABC	IT	1/17/2020	1/14/2021	N
2	124	XYZ	HR	1/18/2020	Null	Y
3	123	ABC	IT	1/15/2021	NULL	Y

Foreign Key

Foreign Key iki tablo arasında relation oluşturmak için kullanılır.
Foreign Key başka bir tablodaki Primary Key ile ilişkilendirilmiş olmalıdır.

Bir tabloda birden fazla Foreign Key olabilir.
Foreign Key NULL değeri 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 Key

Foreign Key tablonun kendi içinde bir ilişki oluşturabilir.

Calisan_ID	Isim	Soy_Isim	Dogum_Tarihi	Cinsiyet	Yillik Maas	Is_ID	Yonetici_ID
1001	Nevzat	Celik	17.07.1995	E	80.000	2	1005
1002	Nurdan	Yilmaz	11.03.1985	K	85.000	3	NULL
2003	Bora	Sverige	06.10.1973	E	78.000	1	1002
1004	Dalyan	Gibiadam	15.07.1983	E	82.000	2	1005
1005	Sükran	Salmaz	11.09.1991	K	90.000	3	NULL

Is_ID	Is_Adi
1	Manual Tester
2	SDET
3	QA Lead

1) Bora Sverige'nin yöneticisi kimdir?
2) Dalyan Gibiadam'in mesleği nedir ?
3) SDET'lerin ortalama maaşı ne kadardır ?
4) En yüksek maaşı alan kişinin mesleği nedir?

SQL Composite Key

Job_ID	Job_Name
2	SDET
3	Manual Tester
1	QE Lead

Job Table

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

Recruiter Table

JobJD	Recruiter 1	
2	Mark Eye	RCG
2	John Ted	RCG
1	Mark Eye	Signature
1	John Ted	Info Log
1	Cory Al	Info Log
2	Angela Star	Signature

Company Table

Composite Key birden fazla field'in kombinasyonu ile oluşturulur.

Tek başına bir field Primary Key olma özelliklerini taşımıyorsa, bu özellikleri elde etmek icin birden fazla field birleştirilerek Primary oluşturulur.

“UNIQUE KEY” & “PRIMARY KEY”

Wise Quarter

“UNIQUE KEY” ve “PRIMARY KEY” arasındaki farklar

Primary Key

Bir Tabloda sadece 1 tane olur.

NULL değer kabul etmez.

Unique Key

Bir tabloda birden fazla olabilir.

Sadece 1 tane NULL değeri kabul eder.

“UNIQUE KEY” ve “PRIMARY KEY” ortak özelliği

Duplication (Çift Kullanım)’a izin vermez.

Örnek Okul Tablosunun Bir Parçası

Wise Quarter

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

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 sahsi 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	

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

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 Çalışma

One to One Relation

Ogrenci_No	Adi	Soy_Adi
101	Abdullah	Keser
102	Adem	Anderson
103	Dalyan	GibiAdam
104	Emine	Yucel
105	Fatih	Bozdemir
106	Jacob	Lejon
107	Levent	Özkul
108	Mehmet	Özkara
109	Talha	Kurt
110	Nese	Özer

Ogrenci_No	Adres	Posta_Kodu	Sehir	Eyalet
101	1453 N 25th St.	34568	Beachwood	Ohio
102	1454 S 6th St.	21385	Avery	Texas
103	1455 E 9th St.	76234	Pittsburgh	Pensylvania
104	1456 NW 32th St.	73173	Baytown	Texas
105	1457 W 3rd St.	98341	Arlington	Virginia
106	1458 SE 1st St.	53485	Miami	Florida
107	1459 SW 17th St.	80294	Orange	California
108	1460 N 45th St.	56341	Hialeah	Florida
109	1461 E 65th St.	80272	Austwell	Texas
110	1462 W 32th St.	56413	Hialeah	Florida

1)Talha Kurt'un adresi nedir?

2) Emine Yücel'in eyaleti nedir?

3) Ögrenci No'su 105 olan kişinin şehri nedir?

Related Tablolarla Çalışma

One to Many Relation

Ders_No	Ders_Adi	Dersin_Kredisi	Ders_Ucreti	Ogretmen_No
100	Matematik	3	500	1
200	Fizik	2	400	2
300	Ingilizce	2	450	3
400	Secmeli	1	350	1

Ogrenci_No	Adi	Soy_Adi	Ders_No
101	Abdullah	Keser	100
102	Adem	Anderson	300
103	Dalyan	GibiAdam	200
104	Emine	Yucel	100
105	Fatih	Bozdemir	400
106	Jacob	Lejon	200
107	Levent	Özkul	100
108	Mehmet	Özkara	300
109	Talha	Kurt	200
110	Nese	Özer	400

1) Matematik dersi alan öğrenciler kimlerdir?
2) Seçmeli ders alan öğrenciler kimlerdir?
3) Ders ücreti 450 olan öğrenciler kimlerdir ?

Related Tablolarla Çalışma

Many to Many Relation

Ogrenci_No	Adi	Soy_Adi
101	Abdullah	Keser
102	Adem	Anderson
103	Dalyan	GibiAdam
104	Emine	Yucel
105	Fatih	Bozdemir
106	Jacob	Lejon
107	Levent	Özkul
108	Mehmet	Özkara
109	Talha	Kurt
110	Nese	Özer

Ogrenci_No	Ogretmen_No
102	1
101	2
102	2
103	1
105	1
107	3
105	1

Ogretmen_No	Adi	Soyadi	Telefon	Bolum
1	Ahmet	Bulutluöz	133746474	Temel Bilimler
2	Ayse	Sezen	123414553	Muhendislik
3	Anil	Sönmez	213425598	Yabanci Dil

1) Öğretmeni Ahmet Bulutluöz olan öğrencilerin isimleri nelerdir?
2) Adem Anderson'in öğretmenlerinin isimleri nelerdir?
3) Fatih Bozdemir'in öğretmenlerinin isimleri nelerdir?

SQL Data Types

String Data Tipi	Açıklama
char (size)	Sabit sayıdaki karakterleri (harf, numara veya özel karakter) tutar. Parantez içindeki boyut uzunluğu belirtir. Char(5) tanımlı bir alana 2 karakterlik bir veri girerseniz 5 byte alan ayırır, yani tanımladığınız boyut kadar. Maksimum 255 karakter barındırabilir. Tekrar eden boşluklar değer alındığı zaman silinir.
varchar (size)	Değişken sayıdaki karakterleri tutar. En fazla 8.000 karaktere kadar depolama yapar. Varchar(5) tanımlı bir alana 2 karakterlik bir veri girerseniz 2 byte alan ayırır. Buradan varchar ile char arasındaki farkı; char tanımlanan boyut kadar alan ayırırken, varchar girilen karakter boyutu kadar alan ayırır şeklinde ifade edebiliriz.
text	En fazla 65.535 karaktere kadar veri girilebilir. (Boşluklar dahildir.)
longtext	4.294.967.295 karaktere kadar metinsel ifadeleri depolayabilir.

SQL Data Types

Numeric Data Tipi	Açıklama
tinyint (size)	Alabileceği değerler –128 ile 127 arasındadır. Unsigned (Sadece pozitif değerler girilecek) olarak tanımlı ise 0 ile 255 arasındadır. “Boyut” ile alabileceği sınırı belirtebiliriz. Hafızada kapladığı alan ise 1 byte
smallint (size)	-32.768 ile 32.767 arasında değer alır. Unsigned tanımlı aralık 0 ile 65535 arasında değer alır. “Boyut” ile alabileceği sınırı belirtebiliriz. Hafızada kapladığı alan: 2 byte
mediumint (size)	-8.388.608 ile 8.388.607 arasında değer alır. Unsigned tanımlı aralık 0 ile 16777215 arasındadır. “Boyut” ile alabileceği sınırı belirtebiliriz. Hafızada kapladığı alan: 3 byte
int	Alabileceği değerler –2147483648 ile 2147483647 arasındadır. Unsigned tanımlı aralık 0 ile 4294967295 arasındadır. “Boyut” ile alabileceği sınırı belirtebiliriz. Hafızada kapladığı alan: 4 byte
bigint (size)	-9.223.372.036.854.775.808 ile 9.223.372.036.854.775.807 arasında değer alır. “Boyut” ile alabileceği sınırı belirtebiliriz. Hafızada kapladığı alan: 8 byte

SQL Data Types

Numeric Data Tipi	Açıklama
float (size,d)	Değer aralıkları; –3.402823466E+38 ile –1.175494351E-38, 0 arası ve 1.175494351E-38 ile 3.402823466E+38 arasındadır. Küçük rakamlı virgüllü ifadeler için kullanılır. “Boyut” ile sayının (virgüllü kısmı dahil) alabileceği en fazla miktar belirtilirken, “d” ile virgülden sonra kaç basamak olacağı belirtilir. Örneğin, 32.658 sayısını saklayacağımız float türü bir sütun tanımlarken, FLOAT(5,3) olarak tanımlarız. Buradaki 5 rakamı, sayının tamamının (noktasız olarak) basamak uzunluğu, 3 rakamı ise noktadan sonraki hane sayısını ifade eder. Float veri türünde eğer ondalıklı hane daha uzun ise sayının yuvarlanma durumu oluşabilir. FLOAT(5,2) olarak tanımladığımız bir sütunda 275.199 sayısını saklamak istersek, MySQL otomatik olarak bu sayıyı 275.20 olarak saklayacaktır. Boyut değeri en fazla 23 olabilir. Unsigned olarak çalışmazlar. Hafızada kapladığı alan: 4 byte.
double (size,d)	Büyük noktalı sayı. Büyük rakamlı virgüllü ifadeler için kullanılır. “Size” ile sayının virgüllü kısmı dahil alabileceği en fazla basamak miktarı belirtilirken, “d” ile virgülden sonra kaç basamak olacağı belirtilir. Boyut değeri en fazla 53 olabilir. Unsigned olarak çalışmazlar. Hafızada kapladığı alan: 8 byte.

SQL Data Types

Numeric Data Tipleri

DBMS and version	*Types*
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)

Date Data Types

SQL Data Types

date: Desteklenen aralık ‘1000-01-01’ ile ‘9999-12-31’ arasıdır. MySQL tarihleri YYYY-MM-DD biçiminde gösterir.

datetime: Desteklenen aralık ‘1000-01-01 00:00:00’ ile ‘9999-12-31 23:59:59’ arasıdır. MySQL DATETIME değerlerini ‘YYYY-MMDD HH:MM:SS’ biçiminde gösterir.

time: Sadece saat verisi saklamak için kullanılır. Desteklenen aralık ‘-838:59:59’ ile ‘838:59:59’ arasındadır. MySQL TIME değerlerini ‘HH:MM:SS’ biçiminde gösterir.

year: 2 veya 4 basamaklı yıl bilgisini saklamak için kullanılır. Dört basamaklı verilerde 1901 ile 2155 arası değer saklanır. İki basamaklı verilerde ise 70 ile 69 (1970 ile 2069) arası değerler saklanır.

Standart “Date Format” , “YYYY- MM - DD”. Örneğin 2020-01-23 Tarih formatını “ALTER SESSION SET NLS_DATE_FORMAT = “YYYY-MMM-DD” kodu kullanılarak değiştirilebilir. Koddan sonra tarih 2020 - Jan - 13 olur.

SQL Data Types

BLOB Data Types

blob

“BLOB” , “Binary Large OBjects” demektir.

“BLOB” resim,video,ses gibi dataları binary formatına çevirerek depolar.

SQL

Structured Query Language

(Yapılandırılmış Sorgu Dili)

3. Ders

Ayşe Sezen

Full-Stack Automation Engineer

BU DERS NELER ÖĞRENECEĞİZ?

1) Primary Key-Foreign Key Tanımlama

2) Tabloya Data Ekleme

SQL Komutları

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

SELECT : Veritabanındaki verileri alır.

2. Veri Değiştirme 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.

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.

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.

Table Nasıl Oluşturulur?

Wise Quarter

1) Create from Scratch

```
CREATE TABLE student_table
(
 id char(11),
 name varchar(50),
 grade int,
 adres varchar(100),
 last_update date
);
```

2) Var olan tablodan yeni tablo oluşturmak

```
CREATE TABLE student_grade
 AS SELECT name,grade
 FROM student_table;
```

Table Nasıl Oluşturulur?

Wise Quarter

1) Create from Scratch

Practice:

"tedarikciler" isminde bir tablo olusturun ve "tedarikci_id", "tedarikci_ismi", "tedarikci_adres" ve "ulasim_tarihi" field'lari olsun.

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

2) Var olan tablodan yeni tablo oluşturmak

"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;
```

Bir field'in "tekrarsız" değer alması nasıl sağlanır?

"id" field'ini "tekrarsiz" yapmak icin , id field'inda Data Type'dan sonra "UNIQUE" yazmak gerekir.

```
CREATE TABLE students_table
(
 id char(11) UNIQUE,
 name varchar(50),
 grade int,
 adres varchar(100),
 last_update date
);
```

Bir Tabloya “Primary Key” Nasil Eklenir

1) Primary Key bir record’u tanimlayan 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’inda hic bir data NULL olamaz.

“id” field’ini “primary key” yapmak icin 2 yol var.

1) Data Type’dan sonra “PRIMARY KEY” yazarak.

```
CREATE TABLE students_table
(
id int PRIMARY KEY,
name varchar(50) NOT NULL,
grade int,
adres varchar(100),
last_update date
);
```

Bir Tabloya “Primary Key” Nasil Eklenir

2) CONSTRAINT Keyword Kullanilarak Primary Key Tanimlanabilir

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

```
CREATE TABLE students
(
id int,
name varchar(50) NOT NULL,
 grade int,
 address varchar(100),last_modification date,
CONSTRAINT id_pk PRIMARY KEY(id)
);
```

Bir Tabloya “Primary Key” Nasil Eklenir

Practice: “sehirler” isimli bir Table olusturun. Tabloda “alan_kodu”, “isim”, “nufus” field’lari olsun. Isim field’i bos birakilamasin. 1.Yontemi kullanarak “alan_kodu” field’ini “Primary Key” yapin

```
CREATE TABLE sehirler
(
alan_kodu int PRIMARY KEY,
name varchar(50) NOT NULL,
isim varchar(20),
nufus int
);
```

Practice: “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” yapin

```
CREATE TABLE ogretmenler
(
id char(10) UNIQUE,
isim varchar(20),
brans varchar(20),
CONSTRAINT ogretmenler_pk
PRIMARY KEY (id,isim)
);
```

Tabloya “Foreign Key” Nasil Eklenir ?

Wise Quarter

- Foreign Key iki tablo arasinda 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 ekleyemezsiniz

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

Tabloya “Foreign Key” Nasil Eklenir ?

Wise Quarter

Primary Key

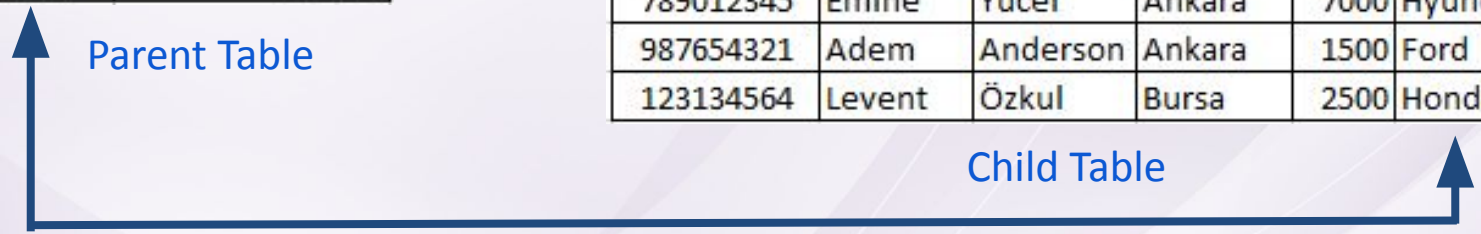
Sirket_id	Sirket	Personel_Sayisi
100	Honda	12000
101	Ford	18000
102	Hyundai	10000
103	Toyota	21000

Parent Table

Foreign Key

ID	Adi	Soy_Adi	Sehir	Maas	Sirket
123456789	Abdullah	Keser	Istanbul	2500	Honda
234567890	Adem	Anderson	istanbul	1500	Toyota
345678901	Emine	Yucel	Ankara	3000	Honda
567890123	Emine	Yucel	Izmir	1000	Ford
789012345	Emine	Yucel	Ankara	7000	Hyundai
987654321	Adem	Anderson	Ankara	1500	Ford
123134564	Levent	Özkul	Bursa	2500	Honda

Child Table



Tabloya “Foreign Key” Nasil Eklenir ?

Practice: “tedarikciler” 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 varchar(50),
iletisim_isim varchar(50),
CONSTRAINT tedarikci_pk
PRIMARY KEY (tedarikci_id)
);
```

Tabloya “Foreign Key” Nasil Eklenir ?

Wise Quarter

Practice:

“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 int,
tedarikci_isim varchar(50),
iletisim_isim varchar(50),
CONSTRAINT tedarikci_pk PRIMARY KEY
(tedarikci_id,tedarikci_isim)
);
```

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