13.HAFTA

Bir data bir de meta data var. Tablonun içinde bir şey olmadan create edilebilmesi mümkün.

**Veri tabanı objelerinin temel tipleri:**

1) TABLO: Verileri gömdüğümüz yer. Veri taşırsa alan işgal eder.

2) INDEX: Tablonun taranması ve search edilmesi ile aakalı

3) CONSTRAİNT: İçinde veri taşımaz, kuralları barındırırlar. Veri tabanına gömdüğümüz kurallardır

4) VİEW: Sadece run ederken tablo gibi karşımıza çıkar.

5) SEQUENCE: Otomatik sıralamaya bağlanan uygulamaları sırayla alması

6) SYNONYM: Bir objenin benzeridir, kopyası.

Fiziksel olarak diskte yer işgal edenler TABLO ve INDEX lerdir.

Constraint aslında veri tabanı kurallarıdır ve bu kurallar data dictionary de yer alır. Örneğin Foreignkey ilişkisi varsa tabloda kontrollerin yapılması için foreign constraintine ihtiyaç vardır

Constrait kullanııosak adını biz vermeliyiz

**DDL KOMUTLARI**

CREATE, ALTER, DROP(ortadan kaldırma), RENAME, TRUNCATE

**CREATE TABLE EXAMPLES**

CREATE TABLE my\_od\_collection

(cd\_number NUMBER(3),

title VARCHAI2(20),

artist VARCHAR2(20),

purchase\_date DATE DEFAULT SYSDATE);

İlişkisel veri tabanında içinde veri olmadan da create edilebilir.

CHAR : Fixed Size

CLOB & VARCHAR2 & NUMBER & RAW & BLOB : Variable Size

DATE TIMESTAMP INTERVAL

**ALTER TABLE(ADD, MODIFY, DROP)**

Yeni sütun ekleme, Var olan sütunun veri tipinde veya size ında değişiklik yapma, sütun güncelleme drop etme… Yeni sütun için Sütun sırasını belirleyemezsin.

Create aşamasında yapmadığımız tanımlamaları yapmamıza işe yarar.

ALTER TABLE tablename

ADD(column name data type[DEFAULT expression],

column name data type[DEFAULT expression]

ALTER TABLE my\_od\_collection

ADD(release date DATE DEFAULT SYSDATE);

ALTER TABLE my friends

ADD(vorite game VARCHAR2(30));

ALTER TABLE mod emp

MODIFY(last\_name VARCHAR2(30));

ALTER TABLE mod\_emp

MODIFY(last\_name VARCHAR2(10));

ALTER TABLE mod\_emp

MODIFY(salary NUMBER(10,2));

ALTER TABLE mod\_emp

MODIFY(salary NUMBER(8,2)DEFAULT 50);

ALTER TABLE my friends

DROP COLUMN favorite\_game;

RENAME my\_od\_collection To my\_music;

ALTER TABLE table\_name

ADD[CONSTRAINT constraint\_name]type of constraint(column\_name);

ALTER TABLE employees

MODIFY(email CONSTRAINT emp\_email\_nn NOT NULL);

**FLASHBACK QUERY**

INSERT INTO copy\_employees

VALUES(1,Natasha','Hansen','NUANSEN','4412312341234',

11-11-1998','AD\_VP',12000,null,100,90,NULL);

SELECT employee\_id,first\_name || || last\_namees"NAME",

versions gesrationAS"OPERATION",

versions starttisAS"START\_DATE",versions entáreas"END\_DATE",salary

FROM copy\_employees

VERSIONS BETWEEN SCN MINVALUE AND MAXVALUE

WHERE employee\_id=1;

UPDATE copy\_employees

SET salary=1

WHERE employee\_id=1;

DELETE from copy\_employees

WHERE employee\_id=1;

CREATE TABLE clients(

client number NUMBER(6),

first\_name VARCHAR2(20),

last\_name VARCHAR2(20),

1 phone VARCHAR2(20) CONSTRAINT phone\_email\_uk UNIQUE(email,phone),

2 email VARCHAR2(10) CONSTRAINT NOT NULL,

3 CONSTRAINT emailclients email NOT NULL,

CONSTRAINT clients\_client\_num\_pk PRIMARY KEY(client\_number));

1 sadece phone olmalı, phone demeye de gerek yok sadece unıq desek de olurdu.

2 Constrainte isim vermemiş sadece not null dese ismi sistem verirdi

3 Tablo bazında hangi sütun olduğunu belirtmemiş.

UNIQUE: O değerin o tabloda tek olmasıdır.

COMPOSİTE UNIQUE: Birlikte değerlendirlir.

NOT NULL Sütun bazında tanımlanır.

PRİMARY KEY birden fazla sütundan oluşursa tablo bazında yazılmak zorunda.

**FOREİGN KEY sütun bazında belirtirsek hangi sütuna reference edilmesi gerektiğini de yazmalıyız;**

CREATE TABLE copy\_employees

(employee\_id NUMBER(6,0) CONSTRAINT copy\_emp\_pk PRIMARY KEY,

first name VARCHAR2(20),

last name VARCHAR2(25),

department\_id NUMBER(4,0) CONSTRAINT c\_emps\_dept\_id\_fk

REFERENCES departments(department\_id),

email VARCHAR2(25));

**FOREİGN KEY tablo bazında belirtirsek FOREIGN KEY yazarak belirtmeliyiz;**

CREATE TABLE copy\_employees

(employee\_id NUMBER(6,0) CONSTRAINT copy\_emp\_pk PRIMARY KEY,

first\_name VARCHAR2(20),

last name VARCHAR2(25),

department\_id NUMBER(4,0),

email VARCHAR2(25),

CONSTRAINT c\_emps\_dept\_id\_fk FOREIGN KEY(department\_id)

REFERENCES departments(department\_id));

**FOREİGN KEY CONSTRAİNT’DE SİLME İŞLEMLİ**

1. **ON DELETE CASCADE**

Çocuk kaydı olduğu için silinemeyeni de silebilmemizi sağlar.

1. **ON DELETE SET NULL**

Çalışanım orda kalsın ama departman sütunum silinsin

**CHECK CONSTRAİNT**

CREATE TABLE copy\_job\_history

(employee\_id NUMBER(6,0),

start\_date DATE,

end date DATE,

job id VARCHAR2(10),

department\_id NUMBER(4,0),

CONSTRAINT ojhist\_ep\_id\_st\_date\_pk

PRIMARY KEY(employee\_id, start\_date),

CONSTRAINT cjhist\_end\_ck CHECK(end\_date>start\_date));

KONTROL SAĞLAMIŞ OLURUZ.

Bazı parametreleri ve call functionları kullanamayız (SYSDATE..)

CURRVAL, NEXTVAL, LEVEL, ROWNUM parametreleri de kullanılamaz.

**Column-level syntax;**

salary NUMBER(8,2) CONSTRAINT employees\_min\_sal\_ck CHECK(salary>0)

**Table-level syntax:**

CONSTRAINT employees\_min\_sal\_ck CHECK(salary>0)

İntegrity Rules için İntegrity Constraintler tanımlanmıştır.

PK, FK, UNIQUE, CHECK Constraintler.