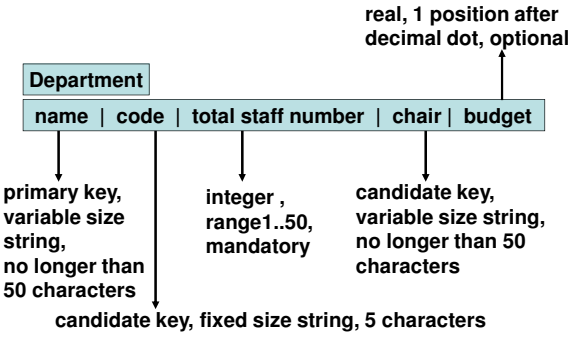
CREATE TABLE statement

Functionality

CREATE TABLE statement creates a new relational table with a given name, given attribute names and types, and with the given logical consistency constraints

Example



Example

```
CREATE TABLE Department(
name                  VARCHAR2(50),
code                  CHAR(5),
total_staff_number    NUMBER(2)       NOT NULL,
chair                 VARCHAR2(50),
budget                NUMBER(9,1)     NULL,
      CONSTRAINT dept_pkey PRIMARY KEY(name),
      CONSTRAINT dept_ckey1 UNIQUE(code),
      CONSTRAINT dept_ckey2 UNIQUE(chair),
      CONSTRAINT dept_check1
      CHECK (total_staff_number BETWEEN 1 AND 50)   );
```


Keywords

```
CREATE TABLE Department(
name                  VARCHAR2(50),
code                  CHAR(5),
total_staff_number    NUMBER(2)       NOT NULL,
chair                 VARCHAR2(50),
budget                NUMBER(9,1),    NULL,
CONSTRAINT dept_pkey PRIMARY KEY(name),
CONSTRAINT dept_ckey1 UNIQUE(code),
CONSTRAINT dept_ckey2 UNIQUE(chair),
CONSTRAINT dept_check1
CHECK (total_staff_number BETWEEN 1 AND 50)   );
```


Table name

```
CREATE TABLE Department(
name                  VARCHAR2(50),
code                  CHAR(5),
total_staff_number    NUMBER(2)       NOT NULL,
chair                 VARCHAR2(50),
budget                NUMBER(9,1)     NULL,
CONSTRAINT dept_pkey PRIMARY KEY(name),
CONSTRAINT dept_ckey1 UNIQUE(code),
CONSTRAINT dept_ckey2 UNIQUE(chair),
CONSTRAINT dept_check1
CHECK (total_staff_number BETWEEN 1 AND 50)   );
```

7. CREATE TABLE

Attribute names

```
CREATE TABLE Department(
name                VARCHAR2(50),
code                CHAR(5),
total_staff_number  NUMBER(2)       NOT NULL,
chair               VARCHAR2(50),
budget              NUMBER(9,1)     NULL,
CONSTRAINT dept_pkey PRIMARY KEY(name),
CONSTRAINT dept_ckey1 UNIQUE(code),
CONSTRAINT dept_ckey2 UNIQUE(chair),
CONSTRAINT dept_check1
CHECK (total_staff_number BETWEEN 1 AND 50)  );
```


7. CREATE TABLE

Attribute types

```
CREATE TABLE Department(
name                VARCHAR2(50),
code                CHAR(5),
total_staff_number  NUMBER(2)       NOT NULL,
chair               VARCHAR2(50),
budget              NUMBER(9,1)     NULL,
CONSTRAINT dept_pkey PRIMARY KEY(name),
CONSTRAINT dept_ckey1 UNIQUE(code),
CONSTRAINT dept_ckey2 UNIQUE(chair),
CONSTRAINT dept_check1
CHECK (total_staff_number BETWEEN 1 AND 50)  );
```


7. CREATE TABLE

Constraints

```
CREATE TABLE Department(
name                VARCHAR2(50),
code                CHAR(5),
total_staff_number  NUMBER(2)       NOT NULL,
chair               VARCHAR2(50),
budget              NUMBER(9,1)     NULL,
CONSTRAINT dept_pkey PRIMARY KEY(name),
CONSTRAINT dept_ckey1 UNIQUE(code),
CONSTRAINT dept_ckey2 UNIQUE(chair),
CONSTRAINT dept_check1
CHECK (total_staff_number BETWEEN 1 AND 50)  );
```


7. CREATE TABLE

Primary key constraint

```
CREATE TABLE Department(
name                VARCHAR2(50),
code                CHAR(5),
total_staff_number  NUMBER(2)       NOT NULL,
chair               VARCHAR2(50),
budget              NUMBER(9,1)     NULL,
CONSTRAINT dept_pkey PRIMARY KEY(name),
CONSTRAINT dept_ckey1 UNIQUE(code),
CONSTRAINT dept_ckey2 UNIQUE(chair),
CONSTRAINT dept_check1
CHECK (total_staff_number BETWEEN 1 AND 50)  );
```


7. CREATE TABLE

Candidate key constraint

```
CREATE TABLE Department(
name                VARCHAR2(50),
code                CHAR(5),
total_staff_number  NUMBER(2)       NOT NULL,
chair               VARCHAR2(50),
budget              NUMBER(9,1)     NULL,
CONSTRAINT dept_pkey PRIMARY KEY(name),
CONSTRAINT dept_ckey1 UNIQUE(code),
CONSTRAINT dept_ckey2 UNIQUE(chair),
CONSTRAINT dept_check1
CHECK (total_staff_number BETWEEN 1 AND 50)  );
```


7. CREATE TABLE

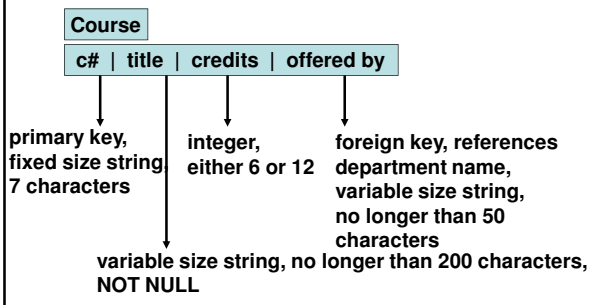
Domain constraint

```
CREATE TABLE Department(
name                VARCHAR2(50),
code                CHAR(5),
total_staff_number  NUMBER(2)       NOT NULL,
chair               VARCHAR2(50),
budget              NUMBER(9,1)     NULL,
CONSTRAINT dept_pkey PRIMARY KEY(name),
CONSTRAINT dept_ckey1 UNIQUE(code),
CONSTRAINT dept_ckey2 UNIQUE(chair),
CONSTRAINT dept_check1
CHECK (total_staff_number BETWEEN 1 AND 50)  );
```

NULL/NOT NULL constraint

```
CREATE TABLE Department(
name                 VARCHAR2(50),
code                 CHAR(5),
total_staff_number   NUMBER(2)       NOT NULL,
chair                VARCHAR2(50),
budget               NUMBER(9,1)     NULL,
CONSTRAINT dept_pkey PRIMARY KEY(name),
CONSTRAINT dept_ckey1 UNIQUE(code),
CONSTRAINT dept_ckey2 UNIQUE(chair),
CONSTRAINT dept_check1
CHECK (total_staff_number BETWEEN 1 AND 50)  );
```


Example



Example

```
CREATE TABLE Course(
c#                CHAR(7),
title             VARCHAR2(200)     NOT NULL,
credits           NUMBER(2)         NOT NULL,
offered_by        VARCHAR2(50)          NULL,
     CONSTRAINT course_pkey PRIMARY KEY(c#),
     CONSTRAINT course_check1
          CHECK (credits IN (6, 12) ),
     CONSTRAINT course_fkey1 FOREIGN KEY(offered_by)
          REFERENCES Department(name)
          ON DELETE CASCADE            );
```


Foreign key constraint

```
CREATE TABLE Course(
c#              CHAR(7),
title           VARCHAR2(200)   NOT NULL,
credits         NUMBER(1)       NOT NULL,
offered_by      VARCHAR2(50)        NULL,
CONSTRAINT course_pkey PRIMARY KEY(c#),
CONSTRAINT course_check1
     CHECK (credits IN (6, 12) ),
CONSTRAINT course_fkey1 FOREIGN KEY(offered_by)
     REFERENCES Department(name)
     ON DELETE CASCADE                    );
```


Another foreign key constraint

```
CREATE TABLE Course(
c#              CHAR(7),
title           VARCHAR2(200)   NOT NULL,
credits         NUMBER(1)       NOT NULL,
offered_by      VARCHAR2(50)        NULL,
CONSTRAINT course_pkey PRIMARY KEY(c#),
CONSTRAINT course_check1
     CHECK (credits IN (6, 12) ),
CONSTRAINT course_fkey1 FOREIGN KEY(offered_by)
     REFERENCES Department(name)
     ON DELETE SET NULL                   );
```


Types of attributes

`VARCHAR2(size)`	Variable length string maximum size `4000` bytes
`CHAR(size)`	Fixed length string maximum size `2000` bytes
`NUMBER(p,s)`	Real number, total `p` digits, `s` digits after decimal dot
`NUMBER(p)`	Integer number, equivalent to `NUMBER(p,0)`
`BLOB`	Variable length sequence of bytes, maximum size `4` Gbytes
`DATE`	Date and time

Types of attributes

NVARCHAR2(size)	Variable length string, national (Unicode-only) character set
NCHAR(size)	Fixed length string, national (Unicode-only) character set
TIMESTAMP	DATE + a fraction of second
RAW	Binary data, maximum size 2000 bytes
LONG RAW	Binary data, maximum size 2 Gbytes
ROWID	Row identifier
BFILE	Reference to a binary file (read only)
XMLType	XML document

ANSI SQL versus Oracle data types

CHARACTER(n)	CHAR(n)
CHARACTER VARYING(n)	VARCHAR(n)
INTEGER	NUMBER(38)
REAL, FLOAT	NUMBER
NUMERIC(p,s)	NUMBER(p,s)
NATIONAL CHARCHER(n)	NCHAR(n)

References

Elmasri R., Navathe S. B., ***Database Systems*****, chapters 4.1, 4.2**

Ramakrishnan R., Gehrke J., ***Database Management Systems*****, chapters 3.1.1, 3.2.1, 3.2.2, 3.3**

https://www.sai.uow.edu.au/oradocs/
SQL Reference, CREATE TABLE statement