

SELECT

```
SELECT [ALL | DISTINCT]
{* | spaltenliste |
funktion}
FROM {tabellenname [AS]
[tabelle alias] |
viewname [AS] [tabelle
alias]} WHERE
auswahlbedingung
[GROUP BY spaltenliste
[HAVING auswahlbedingung] ]
[ORDER BY spalte [ASC |
DESC] [,spalte2 [ASC |
DESC]... ];
```

Bedingung spaltenname

```
Bedingung spaltenname
vergleichsoperator wert
```

Spaltenliste

```
Spaltenliste spaltenname
[[AS] spaltenalias]
[,spaltenname 2 [[AS]
spaltenalias]]...
```

SELECT spaltenliste

```
SELECT spaltenliste
FROM tabelle1
NATURAL JOIN tabelle -
enname2;
```

SELECT spaltenliste

```
SELECT spaltenliste
FROM tabelle1
[tabelle alias]
[[INNER | {LEFT | FULL |
RIGHT [OUTER]}] JOIN
tabelle2 [tabelle
alias] ON tabelle -
enname1.spaltenname =
tabelle2.spalten -
name];
```

SELECT spaltenliste

```
SELECT spaltenliste
FROM tabelle1
{UNION | INTERSECT |
MINUS}
SELECT spaltenliste
FROM tabelle2
```

SELECT spaltenliste

```
SELECT spaltenliste
FROM tabelle1
[[INNER | {LEFT | RIGHT
[OUTER]}] JOIN tabelle -
enname2
USING (spaltenname)];
```

Wertzuweisungen

```
ziel := wert;
SELECT spalte
INTO ziel
FROM .....
[WHERE.....];
```

Tabellen ändern

```
ALTER TABLE tabellenname
{ADD {spaltenname
[constraint]
| DROP {spaltenname
| constraint}
| MODIFY spaltenname
spaltenname
| MODIFY CONSTRAINT
constraint
| RENAME COLUMN
spaltenname TO
Spaltenname};
```

INSERT

```
INSERT INTO tabellenname
(spaltenname1,
spaltenname2... )
VALUES (wert1, wert2... );
```

UPDATE

```
UPDATE tabellenname
SET spaltenname = wert
[, spaltenname2 = wert]
...
[WHERE auswahlbedingung];
```

DELETE

```
DELETE tabellenname
WHERE auswahlbedingung;
```

CREATE TABLE

```
CREATE TABLE
tabellenname(
spaltendefinition
[, spaltendefinition
on2]...
[CONSTRAINT constraint
name
{PRIMARY KEY (spalten
liste)
| FOREIGN KEY (spalten
liste) REFERENCES
tabelle1 (spalten
liste)}]
| CHECK (bedingung))...
```

CREATE [OR REPLACE] VIEW

```
CREATE [OR REPLACE] VIEW
viewname
[(spaltenalias [,
spaltenalias_n]... )]
AS abfrage
```

CREATE SEQUENCE

```
CREATE SEQUENCE
sequenzname
[START WITH anfang -
swert]
[INCREMENT BY wert];
```

CREATE [UNIQUE] INDEX

```
CREATE [UNIQUE] INDEX
indexname
```



By **Kemmojoo**

cheatography.com/kemmojoo/

Published 30th May, 2016.

Last updated 30th May, 2016.

Page 1 of 2.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish Yours!

<https://apollopadd.com>

CREATE [UNIQUE] INDEX (cont)

> ON tabellenname (spaltenname1[ASC | DESC] [, spaltenname2 [ASC | DESC]]...);

CREATE [OR REPLACE] TRIGGER

```
CREATE [OR REPLACE]
TRIGGER triggername
{BEFORE | AFTER}
{DELETE | INSERT |
UPDATE [OF spalte nli -
ste]}... ON tabell enname
FOR EACH ROW
[WHEN (bedin gung)]
[DECLARE
bezeichner datentyp;
...]]
BEGIN anweis ungen;
END;
```

Spaltendefinition

```
spaltenname datentyp
[NOT NULL | DEFAULT
wert]
| UNIQUE | PRIMARY KEY
| REFERENCES tabell -
enname [(spal ten name)]
[ON DELETE CASCADE] |
CHECK (bedin gung)
```

Bedingung Verneinen

NOT

Bedingungen verknüpfen

AND

OR

Datumsfunktionen

CURRENT_DATE

SYSDATE

EXTRACT ({DAY | MONTH | YEAR} FROM datumswert)

Constraint auf Spaltenebene

NULL

NOT NULL

Vergleichsoperatoren

<, <=, >, >=, =, <>

LIKE

IN

IS NULL, IS NOT NULL

BETWEEN wert1 AND wert2

Mathematische Operatoren

* /

+ -

Zeichenketten verbinden

||

Wildcards für LIKE (Oracle, Standard)

%, _

Bedingungen bei Unterabfragen

mit einem Ergebnis: =, !=, <, >, <=, >=

mit mehrzeiligem Ergebnis: IN, EXISTS, ALL, ANY

Constraints auf Tabellenebene

UNIQUE

PRIMARY KEY

FOREIGN KEY

REFERENCES

CHECK

Datentypen (Auszug)

INTEGER, SMALLINT, NUMBER(n1,m), DECIMAL(n,m)

CHAR(anzahl), VARCHAR(anzahl)

DATE, TIMESTAMP

TEXT, CLOB, BLOB

Aggregatfunktionen

SUM(spaltenname)

MIN(spaltenname)

MAX(spaltenname)

AVG(spaltenname)

COUNT({* | ([DISTINCT]spaltenname)})

Zeichenkettenfunktionen

UPPER(spaltenname)

LOWER(spaltenname)

Mathematische Funktionen

MOD(dividend, divisor)

ABS(ausdruck)

POWER(basis,exponent)

SQRT(ausdruck)

ROUND(ausdruck[nachkommastellen])

Konvertierungsfunktionen

TO_CHAR(ausdruck)

TO_NUMBER(ausdruck,formatstring)

TO_DATE(datum,formatstring)

Löschen im Datenbankschema

DROP TABLE tabellenname [CASCADE CONSTRAINTS];

DROP INDEX indexname;

DROP SEQUENCE sequenzname;

DROP TRIGGER triggername;

DROP VIEW viewname;

IF-Anweisung

```
IF bedingung
THEN
anweis ungen;
[ELSE
anweis ungen;]
END IF;
```

Else If-Anweisung

```
ELSIF bedingung
anweis ungen
```

Verhindern von Änderungen mit Fehlermeldung

```
raise_application_error(errornummer2, errortext3);
```