

SELECT

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

Bedingung spaltenname

```
Bedingung spaltenname
vergleichtsoperator wert
```

Spaltenliste

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

SELECT spaltenliste

```
SELECT spaltenliste
FROM tabellenname1
NATURAL JOIN
tabellenname2;
```

SELECT spaltenliste

```
SELECT spaltenliste
FROM tabellenname1
[tabellenalias]
[[INNER | {LEFT | FULL |
RIGHT [OUTER]}] JOIN
tabellenname2
[tabellenalias] ON
tabellenname1.spaltenname =
tabellenname2.spaltenname]
...;
```

SELECT spaltenliste

```
SELECT spaltenliste
FROM tabellenname1 {UNION
| INTERSECT | MINUS}
SELECT spaltenliste
FROM tabellenname2
```

SELECT spaltenliste

```
SELECT spaltenliste
FROM tabellenname1 [[INNER
| {LEFT | RIGHT [OUTER]}]
JOIN tabellenname2
USING (spaltenname)]...;
```

Wertzuweisungen

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

Tabellen ändern

```
ALTER TABLE tabellenname
{ADD {spaltendefinition |
constraint}
| DROP {spaltenname |
constraintname}
| MODIFY spaltenname
spaltendefinition
| MODIFY CONSTRAINT
constraintname
constraintRegel
| RENAME COLUMN
spaltennameAlt TO
SpaltennameNeu };
```

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
[, spaltendefinition2]...
[CONSTRAINT
constraintname
{PRIMARY KEY
(spaltenliste)
| FOREIGN KEY
(spaltenliste) REFERENCES
tabellenname [(spaltenliste
)]
| CHECK (bedingung)}]... );
```

CREATE [OR REPLACE] VIEW

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

CREATE SEQUENCE

```
CREATE SEQUENCE
sequenzname
[START WITH anfangswert]
[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 **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

CREATE [UNIQUE] INDEX (cont) ON tabellenname (spaltenname1[ASC DESC] [, spaltenname2 [ASC DESC]]...);	Bedingung Verneinen NOT	Bedingungen bei Unterabfragen mit einem Ergebnis: =, !=, <, >, <=, >=	Konvertierungsfunktionen TO_CHAR(ausdruck) TO_NUMBER(ausdruck,formatstring) TO_DATE(datum,formatstring)
CREATE [OR REPLACE] TRIGGER CREATE [OR REPLACE] TRIGGER triggername {BEFORE AFTER} {DELETE INSERT UPDATE [OF spaltenliste]}.. ON tabellenname FOR EACH ROW [WHEN (bedingung)] [DECLARE bezeichner datentyp; [...]] BEGIN anweisungen; END;	Bedingungen verknüpfen AND OR	Constraints auf Tabellenebene UNIQUE PRIMARY KEY FOREIGN KEY REFERENCES CHECK	Löschen im Datenbankschema DROP TABLE tabellenname [CASCADE CONSTRAINTS]; DROP INDEX indexname; DROP SEQUENCE sequenzname; DROP TRIGGER triggername; DROP VIEW viewname;
Spaltendefinition spaltenname datentyp [NOT NULL DEFAULT wert] UNIQUE PRIMARY KEY REFERENCES tabellenname [(spaltenname)] [ON DELETE CASCADE] CHECK (bedingung)	Datumsfunktionen CURRENT_DATE SYSDATE EXTRACT ((DAY MONTH YEAR) FROM datumswert)	Datentypen (Auszug) INTEGER, SMALLINT, NUMBER(n1,m), DECIMAL(n,m) CHAR(anzahl), VARCHAR(anzahl) DATE, TIMESTAMP TEXT, CLOB, BLOB	IF-Anweisung IF bedingung THEN anweisungen; [ELSE anweisungen;] END IF;
Constraint auf Spaltenebene NULL NOT NULL	Vergleichsoperatoren <, <=, >, >=, =, <> LIKE IN IS NULL, IS NOT NULL BETWEEN wert1 AND wert2	Aggregatfunktionen SUM(spaltenname) MIN(spaltenname) MAX(spaltenname) AVG(spaltenname) COUNT({* ([DISTINCT]spaltenname)})	Else If-Anweisung ELSIF bedingung anweisungen
Mathematische Operatoren * / + -	Zeichenketten verbinden Wildcards für LIKE (Oracle, Standard) %, _	Zeichenkettenfunktionen UPPER(spaltenname) LOWER(spaltenname)	Verhindern von Änderungen mit Fehlermeldung raise_application_error(errornummer 2, errortext3);
		Mathematische Funktionen MOD(dividend, divisor) ABS(ausdruck) POWER(basis,exponent) SQRT(ausdruck) ROUND(ausdruck[nachkommastell en])	

