

SQL příkazy

Příkazy: [DDL](#)

[DML](#)

[DAS](#)

[DIS](#)

[Embedded SQL commands](#)

[Query Optimization](#)

[Information Stat.](#)

[DATOVÉ TYPY](#)

Syntax [DDL](#)

[DML](#)

[DAS](#)

[DIS](#)

[Embedded SQL commands](#)

[Query Optimization Information](#)

[Stat](#)

PRÍKAZY

DDL

[syntax](#)

ALTER FUNCTION

To recompile a stored function.

ALTER INDEX

To redefine an index's future storage allocation.

ALTER PACKAGE

To recompile a stored package.

ALTER PROCEDURE

To recompile a stored procedure.

ALTER SEQUENCE

To redefine value generation for a sequence.

ALTER SNAPSHOT

To change a snapshot's storage characteristics, automatic refresh time, or automatic refresh mode.

ALTER TABLE

To add a column/integrity constraint to a table.

To redefine a column, to change a table's storage characteristics. To enable/disable/drop an integrity constraint.

To enable/disable tables locks on a table.

To enable/disable all triggers on a table.

To allocate an extent for the table.

To allow/disallow writing to a table.

To modify the degree of parallelism for a table.

ALTER TABLESPACE

To add/rename data files.

To change storage characteristics.

To take a tablespace online/offline.

To begin/end a backup.

To allow/disallow writing to a tablespace.

ALTER TRIGGER

To enable/disable a database trigger.

ALTER USER

To change a user's password, default tablespace, temporary tablespace, tablespace quotas, profile, or default roles.

ALTER VIEW

To recompile a view.

CREATE FUNCTION

To create a stored function.

CREATE INDEX

To create an index for a table or cluster.

CREATE PACKAGE

To create the specification of a stored package.

CREATE PACKAGE BODY

To create the body of a stored package

CREATE PROCEDURE

To create a stored procedure.

CREATE SCHEMA

To issue multiple CREATE TABLE, CREATE VIEW, and GRANT statements in a single transaction.

CREATE SEQUENCE

To create a sequence for generating sequential values.

CREATE SHAPSHOT

To create a snapshot of data from one or more remote master tables.

CREATE SYNONYM

To create a synonym for a schema object.

CREATE TABLE

To create a table, defining its columns, integrity constraints, and storage allocation.

CREATE TRIGGER

To create a database trigger.

CREATE VIEW

To define a view of one or more tables or views.

DROP FUNCTION

To remove a stored function from the database.

DROP INDEX

To remove an index from the database.

DROP PACKAGE

To remove a stored package from the database.

DROP PROCEDURE

To remove a stored procedure from the database.

DROP SEQUENCE

To remove a sequence from the database.

DROP SNAPSHOT

To remove a snapshot from the database.

DROP SYNONYM

To remove a synonym from the database.

DROP TABLE

To remove a table from the database.

DROP TRIGGER

To remove a trigger from the database.

DROP VIEW

To remove a view from the database.

RENAME

To change the name of a schema object.

DML

[*syntax*](#)

DELETE

To remove rows from a table.

INSERT

To add new rows to a table.

SELECT

To select data in rows and columns from one or more tables.

UPDATE

To change data in a table.

Data Access Statements

[*syntax*](#)

GRANT

To grant system privileges, roles and object privileges to users and roles.

LOCK TABLE

To lock a table or view, limiting access to it by other users.

REVOKE

To revoke system privileges, roles, and object privileges from users and roles.

Data integrity Statements

[*syntax*](#)

AUDIT

To choose auditing for specified SQL commands or operations on schema objects.

COMMIT

To make permanent the changes made by statements issued and the beginning of a transaction.

NOAUDIT

To disable auditing by reversing, partially or completely, the effect of a prior AUDIT statement.

ROLLBACK

To undo all changes since the beginning of a transaction or since a savepoint.

SAVEPOINT

To establish a point back to which you may roll.

SET TRANSACTION

To establish properties for the current transaction.

Embedded SQL commands

ALLOCATE

To allocate a cursor variable.

CLOSE

To disable a cursor, releasing the resources it holds.

CONNECT

To log on to an Oracle7 instance.

DECLARE CURSOR

To declare a cursor, associating it with a query.

DECLARE DATABASE

To declare the name of a remote database.

DECLARE STATEMENT

To assign a SQL variable name to a SQL statement.

DECLARE TABLE

To declare the structure of a table for semantic checking of embedded SQL statements by the Oracle Precompiler.

DESCRIBE

To initialize a descriptor, a structure holding host variable descriptions.

EXECUTE

To execute a prepared SQL statement or PL/SQL block or to execute an anonymous PL/SQL block.

EXECUTE IMMEDIATE

To prepare and execute a SQL statement containing no host variables.

FETCH

To retrieve rows selected by a query.

OPEN

To execute the query associated with a cursor.

PREPARE

To parse a SQL statement.

TYPE

To perform user-defined equivalencing.

VAR

To perform host variable equivalencing.

WHENEVER

To specify handling for error and warning conditions.

Query Optimization Information Stat.

[syntax](#)

EXPLAIN PLAN

To explain access path to data

ANALYZE

To collect performance statistics, validate structure, or identify chained rows for a table, cluster, or index.

DATOVÉ TYPY

CHAR (size)

Description : Fixed-length character data of length size.

Column Length (bytes) : Fixed for every row in the table (with trailing blanks); maximum size is 255 bytes per row, default size is one byte per row. Consider the character set that is used before setting size. (Are you using a one-byte or multi-byte character set?)

VARCHAR2 (size)

Description : Variable-length character data.

Column Length (bytes) : A maximum size must be specified. Variable for each row, up to 2000 bytes per row. Consider the character set that is used before setting size. (Are you using a one-byte or multi-byte character set?)

NUMBER (p, s)

Description : Variable-length numeric data.

Column Length (bytes) : Maximum precision p and/or scale s is 38. Variable for each row. The maximum space required for a given column is 21 bytes per row.

DATE

Description : Fixed-length date and time data, ranging from January 1, 4712 B.C. to December 31, 4712 A.D. Default format: DD-MON-YY.

Column Length (bytes) : Fixed at seven bytes for each row in the table.

LONG

Description : Variable-length character data.

Column Length (bytes) : Variable for each row in the table, up to $2^{31} - 1$ bytes, or two gigabytes, per row.

RAW (size)

Description : Variable-length raw binary data.

Column Length (bytes) : A maximum size must be specified. Variable for each row in the table, up to 255 bytes per row.

LONG RAW

Description : Variable-length raw binary data.

Column Length (bytes) : Variable for each row in the table, up to $2^{31} - 1$ bytes, or two gigabytes, per row.

ROWID

Description : Binary data representing row addresses.

Column Length (bytes) : Fixed at six bytes for each row in the table.

MLSLABEL

Description : Variable-length binary data representing operating system labels.

Column Length (bytes) : Variable for each row in the table, ranging from two to five bytes per row.

Syntax jazyka:

Pravidlá popisu:

Kľúčové slová sú uvedené veľkými písmenami
Malými písmenami sú písané parametre
[] označujú nepovinnú voľbu
{ } označujú výber z viacerých možností
| oddeľovač viacerých možností
podčiarknutie označuje (default) prednastavenú hodnotu

Syntax - DDL

ALTER FUNCTION [schema.] function COMPILE

ALTER INDEX [schema.] index

```
[
    PCTFREE integer    |    INITRANS integer    |    MAXTRANS integer    |    STORAGE storage\_clause
    |
    ALLOCATE EXTENT
    [ ( { SIZE integer [K | M] | DATAFILE 'filename' | INSTANCE integer } [...] ) ]
]
[
    DEALLOCATE UNUSED  [KEEP integer [K | M] ]
    |
    REBUILD
    [ {PARALLEL integer | NOPARALLEL} | {RECOVERABLE | UNRECOVERABLE} | TABLESPACE
    tablespace] [...]
]
```

ALTER PACKAGE [schema.] package COMPILE
[PACKAGE | BODY]

ALTER PROCEDURE [schema.] procedure COMPILE

ALTER SEQUENCE [schema.] sequence

```
{
    INCREMENT BY integer
    |
    {MAXVALUE integer | NOMAXVALUE} | {MINVALUE integer | NOMINVALUE}
    |
    {CYCLE | NOCYCLE} | {CACHE integer | NOCACHE} | {ORDER | NOORDER}
} [...]
```

ALTER SNAPSHOT [schema.] snapshot

```
[
    PCTFREE integer | PCTUSED integer | INITRANS integer | MAXTRANS integer | STORAGE
    storage\_clause
] [...]
[ USING INDEX
    [ PCTFREE integer | PCTUSED integer | INITRANS integer | MAXTRANS integer |
    STORAGE storage\_clause
    ]
]
[ REFRESH [FAST | COMPLETE | FORCE] [START WITH date] [NEXT date]]
```

ALTER TABLE [schema.] table

```
{
    ADD ( {column datatype [DEFAULT expr] [column\_constraint [...]]
    |
    table\_constraint
    } [...])
    |
    MODIFY ( { column [datatype] [DEFAULT expr] [column\_constraint [...]] } [...])
    |
    PCTFREE integer | PCTUSED integer | INITRANS integer | MAXTRANS integer | STORAGE storage\_clause
    |
    DROP drop\_clause
}
```

```

ALLOCATE EXTENT [ ( {SIZE integer [K | M] | DATAFILE 'filename' | INSTANCE integer } [...] ) ]
|
DEALLOCATE UNUSED [KEEP integer [K | M]]
}[...]
[ ENABLE {enable\_clause | TABLE LOCK } | DISABLE {disable\_clause | TABLE LOCK } ] [...]
[PARALLEL parallel\_clause] {NOCACHE | CACHE}

```

```

ALTER TRIGGER [schema.] trigger
{ ENABLE | DISABLE | COMPILE }

```

```

ALTER USER user
{
  IDENTIFIED {BY password | EXTERNALLY}
|
  DEFAULT TABLESPACE tablespace
|
  TEMPORARY TABLESPACE tablespace
|
  { QUOTA {integer [K | M] | UNLIMITED } ON tablespace } [...]
|
  PROFILE profile
|
  DEFAULT ROLE {role [,...] | ALL [EXCEPT role [,...]] | NONE }
} [...]

```

```

ALTER VIEW [schema.] view COMPILE

```

```

ANALYZE
{ INDEX | TABLE | CLUSTER } [schema.] { index | table | cluster }
{ COMPUTE STATISTICS [FOR for\_clause]
|
  ESTIMATE STATISTICS | [FOR for\_clause] [SAMPLE integer {ROWS | PERCENT}]
|
  DELETE STATISTICS | VALIDATE STRUCTURE [CASCADE] | LIST CHAINED ROWS [INTO [schema.]
table]
}

```

```

CREATE [OR REPLACE] FUNCTION [schema.] function
[ ( { argument [IN | OUT | IN OUT] datatype } [,...]) ]
RETURN datatype { IS | AS } pl/sql_subprogram_body

```

```

CREATE [UNIQUE] INDEX [schema.] index ON
{
  [schema.] table ( { column [ASC | DESC] } [,...])
|
  CLUSTER [ schema. ] cluster
}
[ INITRANS integer | MAXTRANS integer | TABLESPACE tablespace | STORAGE storage\_clause
| PCTFREE integer | NOSORT | {RECOVERABLE | UNRECOVERABLE }
] [...]
[PARALLEL parallel\_clause]

```

```

CREATE [OR REPLACE] PACKAGE [schema.] package
{IS | AS } pl/sql_package_spec

```

```

CREATE [OR REPLACE] PACKAGE BODY [schema.] package
{IS | AS } pl/sql_package_body

```

```

CREATE [OR REPLACE] PROCEDURE [schema.] procedure
[ ( { argument [IN | OUT | IN OUT] datatype } [,...]) ]
{IS | AS } pl/sql_subprogram_body

```

```

CREATE SCHEMA AUTHORIZATION schema
{ CREATE VIEW command | GRANT command | CREATE TABLE command } [...]

```

```
CREATE SEQUENCE [schema.] sequence
[ START WITH integer | INCREMENT BY integer
|
{MAXVALUE integer | NOMAXVALUE} | {MINVALUE integer | NOMINVALUE}
| {CYCLE | NOCYCLE} | {CACHE integer | NOCACHE} | {ORDER | NOORDER}
] [...]
```

```
CREATE SNAPSHOT [schema.] snapshot
{
{ PCTFREE integer | PCTUSED integer | INITRANS integer
|
MAXTRANS integer | TABLESPACE tablespace | STORAGE storage\_clause
}
|
CLUSTER cluster ( column [...])
} [...]
[ USING INDEX
[ PCTFREE integer | PCTUSED integer | INITRANS integer | MAXTRANS integer ]
]
[ REFRESH [FAST | COMPLETE | FORCE] [START WITH date] [NEXT date] ]
[ FOR UPDATE] AS subquery
```

```
CREATE [PUBLIC] SYNONYM [schema.] synonym
FOR [schema.] object [@dblink]
```

```
CREATE TABLE [schema.] table
[ ( { column datatype [DEFAULT expr] { column\_constraint } [...]
|
table\_constraint
} [...] ) ]
[ PCTFREE integer | PCTUSED integer | INITRANS integer
|
MAXTRANS integer | TABLESPACE tablespace | STORAGE storage\_clause
|
{RECOVERABLE | UNRECOVERABLE}
] [...]
|
{CLUSTER cluster (column [...]) }
[PARALLEL parallel\_clause]
{ [ENABLE enable\_clause] | DISABLE disable\_clause } [...]
[AS subquery]
[CACHE | NOCACHE]
```

```
CREATE [OR REPLACE] TRIGGER [schema.] trigger
{BEFORE | AFTER}
{DELETE | INSERT | UPDATE [ OF column [...] ] } [ ... OR ...]
ON [schema.] table
[ REFERENCING { OLD [AS] old | NEW [AS] new } [...] ]
[FOR EACH ROW [WHEN (condition)]]
pl/sql_block
```

```
CREATE [OR REPLACE] [FORCE | NO FORCE] VIEW [schema.] view
[ (alias [...]) ] AS subquery
[WITH [READ ONLY | CHECK OPTION [CONSTRAINT constraint]] ]
```

```
DROP FUNCTION [schema.] function
```

```
DROP INDEX [schema.] index
```

```
DROP PACKAGE [BODY] [schema.] package
```

```
DROP PROCEDURE [schema.] procedure
```

```
DROP SEQUENCE [schema.] sequence
```

```
DROP SNAPSHOT [schema.] snapshot
```

```
DROP [PUBLIC] SYNONYM [schema.] synonym
```

```
DROP TABLE [schema.] table [CASCADE CONSTRAINTS]
```

DROP TRIGGER [schema.] trigger

DROP USER user [CASCADE]

DROP VIEW [schema.] view

RENAME old TO new

Syntax - DML

DELETE [FROM]
 { [schema.] {table | view} [@dblink] } | (subquery) [alias]
 [WHERE condition]

INSERT INTO {[schema.]{table | view } [@dblink] } | (subquery_1)
 [(column [...])]
 { VALUES (expr [...]) | subquery_2 }

SELECT [ALL | DISTINCT] { * | [column_expression [new_name]] [...]}
 FROM meno_tabulky [alias_tabulka] [...]
 [WHERE podmienka]
 [GROUP BY zoznam_stĺpcov] [HAVING podmienka]
 [ORDER BY meno_stĺpca [ASC | DESC] [...]

UPDATE
 [schema.]{ table | view | snapshot } [@dblink] | (subquery_1) [t_alias]
 SET
 { (column [...]) = (subquery_2)
 |
 column = { expr | (subquery_3) } } [...]
 [WHERE condition]

Syntax - DAS

GRANT { system_priv | role } [...] TO {user | role | PUBLIC } [...]
 [WITH ADMIN OPTION]

GRANT { object_priv | ALL [PRIVILEGES] [(column[,...])] } [...]
 ON [schema.] object
 TO {user | role | PUBLIC} [...]
 [WITH GRANT OPTION]

LOCK TABLE { [schema.] {table | view } [@dblink] }
 IN lockmode MODE [NOWAIT]

REVOKE { system_priv | role } [...]
 FROM {user | role | PUBLIC} [...]

REVOKE { object_priv | ALL [PRIVILEGES] } [...]
 ON [schema.] object
 FROM { user | role | PUBLIC } [...]
 [CASCADE CONSTRAINTS]

Syntax - DIS

AUDIT {statement_opt | system_priv } [...]
 [BY user [...]]
 [BY {SESSION | ACCESS}]
 [WHENEVER [NOT] SUCCESSFUL]

AUDIT {object_opt}{...} ON
 { [schema.] object | DEFAULT }
 [BY {SESSION | ACCE}]
 [WHENEVER [NOT] SUCCESSFUL]

COMMIT
 [WORK]
 [COMMENT 'text' | FORCE 'text' [, integer]]

NOAUDIT {statement_opt | system_priv} [...][BY user [...]]
 [WITH GRANT OPTION [NOT] SUCCESSFUL]

NOAUDIT {object_opt}{...} ON [schema.] object
 [WHENEVER [NOT] SUCCESSFUL]

ROLLBACK
 [WORK]
 [TO [SAVEPOINT] savepoint | FORCE 'text']

SAVEPOINT savepoint

Query Optimization Information Stat.

ANALYZE
 { INDEX | TABLE | CLUSTER } [schema.] { index | table | cluster }
 { COMPUTE STATISTICS [FOR *for_clause*]
 |
 ESTIMATE STATISTICS [FOR *for_clause*] [SAMPLE integer {ROWS | PERCENT}]
 |
 DELETE STATISTICS
 |
 VALIDATE STRUCTURE [CASCADE]
 |
 LIST CHAINED ROWS [INTO [schema.] table]
 }

EXPLAIN PLAN [SET STATEMENT_ID = 'text']
 [INTO [schema.] table [:@dblink]]
 FOR statement

SET TRANSACTION
 { READ ONLY | READ WRITE
 |
 ISOLATION LEVEL {SERIALIZABLE | READ COMMITTED}
 |
 USE ROLLBACK SEGMENT rollback_segment
 }

KAUZULY

ARCHIVE_LOG_CLAUSE

ARCHIVE LOG [THREAD integer]
{ {
 SEQUENCE integer
 |
 CHANGE integer
 |
 CURRENT GROUP integer
 |
 LOGFILE 'filename' NEXT
 |
 ALL
 |
 }


```

        START
    } [TO 'location']
    |
    STOP
}

```

CONSTRAINT_CLAUSE

```

table_constraint ::=
[CONSTRAINT constraint]
{
    { UNIQUE | PRIMARY KEY } ( { column } [...])
    |
    FOREIGN KEY ( { column } [...]) REFERENCES [schema.] table [ ( { column } [...]) ] [ON DELETE
CASCADE ]
    |
    CHECK (condition)
}
{
    [ USING INDEX
    [
        PCTFREE integer
        |
        INITRANS integer
        |
        MAXTRANS integer
        |
        TABLESPACE tablespace
        |
        STORAGE storage_clause NOSORT
        |
        { RECOVERABLE | UNRECOVERABLE }
    ] [...]
    |
    DISABLE
    ]
}
}

```

```

column_constraint ::=
[CONSTRAINT constraint]
{
    [NOT] NULL
    |
    {UNIQUE | PRIMARY KEY}
    |
    REFERENCES [schema.] table [ ( column ) ] [ON DELETE CASCADE ]
    |
    CHECK (condition)
}
{
    [ USING INDEX
    [
        PCTFREE integer
        |
        INITRANS integer
        |
        MAXTRANS integer
        |
        TABLESPACE tablespace
        |
        STORAGE storage_clause NOSORT
        |
        { RECOVERABLE | UNRECOVERABLE }
    ] [...]
    |
    DISABLE
    ]
}
}

```

DEALLOCATE_CLAUSE

```

DEALLOCATE UNUSED [KEEP integer [K | M]]

```

DISABLE Clause

```
DISABLE
{
    { UNIQUE (column [,...])
      | PRIMARY KEY
      | CONSTRAINT constraint
    } [CASCADE]
    | ALL TRIGGERS
}
```

DROP Clause

```
DROP
{
    { PRIMARY | UNIQUE (column [,...]) } [CASCADE]
    | CONSTRAINT constraint
}
```

ENABLE Clause

```
ENABLE
{
    {
        { UNIQUE (column [,...])
          | PRIMARY KEY
          | CONSTRAINT constraint
        }
        [USING INDEX
        [ INTRANS integer
          | MAXTRANS integer
          | TABLESPACE tablespace
          | STORAGE storage_clause
          | PCTFREE integer
        ]
        ]
    }
    | ALL TRIGGERS
}
```

FILESPEC (DATA FILES) ::=
'filename' [SIZE integer [K | M]] [REUSE]

FILESPEC (REDO LOG FILE GROUPS) ::=
{ 'filename' | ({ filename } [,...]) } [SIZE integer [K | M]] [REUSE]

FOR Clause ::=
[FOR TABLE
| FOR ALL [INDEXED] COLUMNS [SIZE integer]
| FOR COLUMNS [SIZE integer] {column [SIZE integer]} [...]
| FOR ALL INDEXES
]

PARALLEL Clause
{ NOPARALLEL

```

|
PARALLEL
  ( { DEGREE {integer | DEFAULT}
    |
      INSTANCES {integer | DEFAULT}
    } [...] )
}

```

RECOVER Clause

```

RECOVER [AUTOMATIC] [FROM 'location']
{ {
  { [[STANDBY] DATABASE]}
  [ UNTIL CANCEL
    |
      UNTIL TIME date
    |
      UNTIL CHANGE integer
    |
      USING BACKUP CONTROLFILE
  ] [...]
}
|
TABLESPACE {tablespace [...]}
|
DATABASE {'filename' [...]}
|
LOGFILE 'filename'
|
CONTINUE [DEFAULT]
|
CANCEL
}
[PARALLEL parallel_clause]

```

STORAGE Clause

```

STORAGE
( {
  INITIAL integer [K | M]
  |
  NEXT integer [K | M]
  |
  MINEXTENTS integer
  |
  MAXEXTENTS { integer | UNLIMITED }
  |
  PCTINCREASE integer
  |
  FREELISTS integer
  |
  FREELIST GROUPS integer
  |
  OPTIMAL { integer [K | M] | NULL }
} [...]
)

```