

# Databázové systémy a SQL

## Lekce 4

Daniel Klimeš

## • Nejvýznamnější databázové objekty

- Tabulky (tables)
- Pohledy (views)
- Indexy (indexes)
- Sekvence (sequences)
- Procedury (procedures)
- Funkce (functions)
- Triggery (triggers)

Informace o objektech jsou uloženy v metadatech (systémových datech) databáze  
Přístup k nim je databázově specifický

## ORACLE

- systémové tabulky – uživatelům pouze pro čtení
- metatabulka o metatabulkách – DICTIONARY
- tabulky USER\_XXX – objekty vytvořené uživatelem
- tabulky ALL\_XXX – objekty přístupné uživateli
- tabulky DBA\_XXX – všechny objekty databáze – přístupné jen administrátorovi

- Metatabulky USER\_TABLES, ALL\_TABLES, DBA\_TABLES
  - sloupec table\_name
- metatabulka TAB
  - sloupec tname, tabtype
- Sloupce tabulky - USER\_TAB\_COLUMNS
  - table\_name, column\_name, data\_type

DDL příkazy pro manipulaci s tabulkami

- CREATE TABLE
- DROP TABLE
- ALTER TABLE
- RENAME TABLE

- Pohled = uložený SQL dotaz
- Pracuje se s ním stejně jako s tabulkou
- Ve většině případů je možný pouze SELECT
- CREATE VIEW **v\_ukazka** AS  
SELECT ps.patient\_id, study\_name FROM patient\_study ps, studies s  
WHERE ps.study\_id = s.study\_id

```
SELECT study_name, count(*) FROM v_ukazka
GROUP BY study_name
```

DDL pro pohledy:

```
CREATE OR REPLACE VIEW AS
DROP VIEW
```

ORACLE metadata

- user\_views, tab

- Indexy jsou obdobou kartotéky
- Umožňují rychlejší vyhledávání záznamů ve velkých tabulkách
- Urychlují SELECT dotazy, zpomalují INSERT, UPDATE, DELETE



- Indexy se vytváří nad jedním nebo více sloupci tabulky
- Standardně nad primárním klíčem a cizími klíči
- Dále nad sloupci, které se často používají za WHERE

- DDL pro indexy
  - CREATE INDEX
  - DROP INDEX
  - ALTER INDEX

ORACLE metadata  
user\_indexes  
ind

- Sekvence generují za všech okolností unikátní čísla – posloupnost
  - Použití pro primární klíče při insertech nových řádků
  - `SELECT jmeno_sekv.NEXTVAL from DUAL`
  - `SELECT jmeno_sekv.CURRVAL from DUAL`
  - Každé zavolání `NEXTVAL` vrátí další číslo v posloupnosti bez ohledu na transakce
- Při neúspěšném použití vygenerovaného ID vznikají “díry” v posloupnosti

- ORACLE DDL

- `CREATE SEQUENCE`
- `DROP SEQUENCE`
- `ALTER SEQUENCE`

- ORACLE metadata

- `user_sequences`

- FIREBIRD

- `CREATE GENERATOR`
- `DROP GENERATOR`

1) Zjistěte počet řádků v tabulce questions

```
SELECT COUNT(*) FROM questions
```

2) Ověřte, zda QUESTION\_ID je unikátní v tabulce questions

```
SELECT COUNT(*), COUNT(DISTINCT question_id) FROM questions
```

3) Ověřte, zda QUESTION\_DESCRIPTION je unikátní

```
SELECT COUNT(*), COUNT(DISTINCT question_description) FROM questions
```

4) Vypište QUESTION\_DESCRIPTION, které se opakují více než 10x

```
SELECT question_description, COUNT(*) FROM questions
GROUP BY question_description
HAVING COUNT(*) > 10
```

Najděte formulář s největším počtem otázek

- tabulky CLUSTERS
  - QUESTION\_GROUP\_CLUSTERS
  - QUESTION\_GROUP\_QUESTIONS
- Vypište vše z uvedených tabulek vnitřním spojením přes příslušné klíče

```
SELECT *
FROM clusters c,
      question_group_clusters qg_cluster,
      question_group_questions qg_question
WHERE c.cluster_id = qg_cluster.cluster_id
AND qg_cluster.question_group_id = qg_question.question_group_id
```



- Seskupte dle cluster\_id a cluster\_description a spočítejte počet řádků  
= počet otázek

```
SELECT c.cluster_id, c.cluster_description, COUNT(*)
FROM clusters c,
question_group_clusters qg_cluster,
question_group_questions qg_question
WHERE c.cluster_id = qg_cluster.cluster_id
AND qg_cluster.question_group_id = qg_question.question_group_id
GROUP BY c.cluster_id, c.cluster_description
```

- Vypište řádky s maximální hodnotou COUNT(\*)

```

SELECT cluster_description, pocet FROM (
SELECT c.cluster_id, c.cluster_description, COUNT(*) pocet
FROM clusters c,
question_group_clusters qg_cluster,
question_group_questions qg_question
WHERE c.cluster_id = qg_cluster.cluster_id
AND qg_cluster.question_group_id = qg_question.question_group_id
GROUP BY c.cluster_id, c.cluster_description
ORDER BY count(*) DESC
) WHERE ROWNUM = 1

```

- Najděte záznamy v tabulce QUESTIONS s datatype = 'E', pro které neexistuje záznam v tabulce DISCRETE\_VALUE\_GROUPS

```
SELECT * FROM questions q WHERE datatype ='E'
AND NOT EXISTS(SELECT * FROM discrete_value_groups dvg
WHERE Q.DISCRETE_VALUE_GROUP_ID = DVG.DISCRETE_VALUE_GRP_ID)
```

## Tabulka Pacienti

- ID
- Jmeno
- Datum\_narozeni
- Pohlavi

## Tabulka Vysetreni

- ID\_vysetreni
- ID
- Datum\_vysetreni
- Typ\_vysetreni
- Vysledek



## Minulé cvičení:

1. Zjistěte počet pacientů
2. Zjistěte počet vyšetření typu 7
3. Vypište přehled  
jmeno, počet vyšetření

1. `SELECT COUNT(*) FROM pacienti`
2. `SELECT COUNT(id_vysetreni) FROM vysetreni WHERE typ_vysetreni = 7`
3. `SELECT p.id, p.jmeno, COUNT(id_vysetreni) AS pocet_vysetreni  
FROM pacienti p LEFT JOIN vysetreni v ON p.id = v.id  
GROUP BY p.id, p.jmeno`

## Tabulka Pacienti

- ID
- Jmeno
- Datum\_narozeni
- Pohlavi

## Tabulka Vysetreni

- ID\_vysetreni
- ID
- Datum\_vysetreni
- Typ\_vysetreni
- Vysledek



## Cvičení:

1. Ověřte, zda ID\_vysetreni v tabulce vysetreni je unikátní
2. Vypište id\_vysetreni z tabulky vysetreni, která jsou duplicitní
3. Ověřte, zda každé vyšetření má rodičovský záznam v tabulce pacienti