

- **Zobrazte všetkých hercov, ktorých priezvisko začína na písmeno T.**

- jednoduchý dotaz, stačí vedieť, že na výskyt reťazca používame výraz LIKE a na vyjadrenie ľubovoľného počtu ľubovoľných znakov (písmen) znak %
- v dotaze – last\_name začína na T a za ním ide ľubovoľný text

```
SELECT * FROM actor
WHERE last_name LIKE "T%"
```

- **Zobrazte názvy filmov, kde replacement\_cost > 10 a < 12,50 (tab. film)**

- len spojíme dve podmienky, pričom ich súčasnú platnosť vyjadríme pomocou AND, v oboch častiach podmienky musí byť uvedený názov stĺpca – rovnako ako v prípade premenných pri programovaní
- ako desatinný oddeľovač používame bodku

```
SELECT * FROM film
WHERE replacement_cost > 10 AND replacement_cost < 12.5
```

- **Aká je priemerná dĺžka filmu a koľko trvá najkratší film? (tab.film)**

- obe úlohy si vyžadujú použitie agregáčnych funkcií, ktoré robia výpočty cez všetky záznamy tabuľky – v tomto prípade AVG a MIN
- vzhľadom na to, že obe dávajú ako výsledok jedinú hodnotu, môžeme ich do dotazu napísať naraz
- za účelom zrozumiteľnejšieho pomenovania vo výsledku používame AS na pomenovanie stĺpca s výsledkom
- pozor, medzi agregáčnou funkciou a jej zátvorkou nesmie byť medzera

```
SELECT AVG(length) AS priemer, MIN(length) AS najkratsi FROM film
```

- **Prípravte abecedný zoznam zákazníkov obchodu s id = 2 (tab. customer)**

- jednoduchý dotaz s podmienkou a zoradením, zoradenie je vyjadrené cez ORDER BY názov, pričom pokiaľ chceme usporiadať od najväčšieho k najmenšiemu, použijeme za názvom poľa DESC

```
SELECT * FROM customer
WHERE store_id = 2
ORDER BY last_name
```

- **Koľko zákazníkov je neaktívnych? (tab. customer)**

- kontrola na obsah položky Active – obsahuje hodnoty 0 a 1, neaktívni budú zrejme zákazníci s 0, pre zistenie ich počtu použijeme agregáčnú funkciu COUNT

```
SELECT COUNT(customer_id) AS pocet FROM customer
WHERE active = 0
```

- **Koľkým zákazníkom končí priezvisko na „son“? (tab. customer)**

- opäť s využitím LIKE vyfiltrujeme zákazníkov a pomocou COUNT zistíme počet

```
SELECT COUNT(customer_id) AS pocet FROM customer
```

```
WHERE last_name LIKE "%son"
```

- **Kedy bola uskutočnená prvá platba za výpožičku? (tab. payment)**

- v preklade do nášho jazyka: aký bol najmenší (minimálny) dátum, keď prebehla prvá platba?

```
SELECT MIN(payment_date) AS min_datum FROM payment
```

- dotaz vráti najmenšiu hodnotu daného poľa, v ktorej je aj čas; pokiaľ nás zaujíma z údaju len dátum, použijeme naň funkciu DATE()

```
SELECT DATE(MIN(payment_date)) AS min_datum FROM payment
```

- **Koľko zaplatili zákazníci spolu v roku 2005 za požičanie filmov? (tab. payment)**

- sčítame SUM zaplatené čiastky (amount) z roku 2005
- rok z dátumu získame pomocou funkcie YEAR, alebo zápisom, že dátum je väčší ako 31.12.2004 a menší ako 1.1.2006

```
SELECT SUM(amount) AS platby FROM payment  
WHERE YEAR(payment_date) = 2005
```

alebo

```
SELECT SUM(amount) AS platby FROM payment  
WHERE payment_date > "2004-12-31" AND payment_date < "2006-01-01"
```

- **Koľko rôznych zákazníkov zrealizovalo platbu? (tab. payment)**

- počet platieb by sme dokázali zistiť pomocou COUNT, počet jedinečných záznamov s menom pomocou DISTINCT – kombinácia vyrieši zadanie – zistíme počet jedinečných záznamov s menom

```
SELECT COUNT(DISTINCT customer_id) AS pocet FROM payment
```

- **Vypíšte také krstné mená zákazníkov, ktoré sa v tab. customer nachádzajú viac ako jeden krát.**

- prvá zaujímavejšia úloha – potrebujeme vytvoriť skupiny záznamov na základe mien – GROUP BY meno a zistiť, koľko má ktorá z nich položiek COUNT
- z výsledku zobrazíť len tie mená, kde počet prekročil hodnotu 1
- vzhľadom na to, že filtrujeme výsledok získaný agregáčnou funkciou, nepostačí nám použiť WHERE, ale HAVING

```
SELECT first_name, COUNT(first_name) AS pocet FROM customer  
GROUP BY first_name  
HAVING pocet > 1
```

- **Vypíšte štát a počet miest, ktoré sú v db evidované od najväčšieho k najmenšiemu. (tab. city, resp. aj tab. country)**

- pre každý štát vypíšeme počet miest v ňom – urobíme skupiny po štátoch a zistíme početnosť záznamov v nich
- tieto údaje získame spojením tabuliek (INNER JOIN) country a city cez pole country\_id
- výsledok usporiadame podľa získanej početnosti – ORDER funguje aj nad výsledkom agregácie a pomocou DESC usporiadame od najväčšieho

```
SELECT country.country, COUNT(city.city) as pocet FROM country
INNER JOIN city ON (city.country_id = country.country_id)
GROUP BY country.country
ORDER BY pocet DESC
```

- **Uveďte názov filmu a dĺžku jeho trvania v hodinách (na 2 desatinné miesta)**
  - výpočty môžeme realizovať nad celým stĺpcom tabuľky alebo v rámci záznamu; všetky funkcie okrem agregáčnych pracujú len v rámci záznamu
  - nad záznamom možno použiť aj numerické operácie napr. „/“ pre vydelenie dĺžky v minútach na hodiny
  - získaný výsledok pomocou funkcie ROUND zaokrúhlime na počet miest, ktorý je zadaný ako druhý argument

```
SELECT title, ROUND(length/60,2) AS dlzka FROM film
```

- **Aký najväčší počet znakov má názov niektorého z filmov uvedených v db?**
  - funkcia pre zistenie počtu znakov je LENGTH, na zistenie najdlhšieho filmu by postačilo použiť MAX (LENGTH), ale pokiaľ nás zaujíma aj názov filmu, usporiadame ich podľa dĺžky názvu
  - pomocou LIMIT určíme koľko záznamov od začiatku chceme zobraziť
  - pozor, nefungovalo by to v prípade, ak by malo najdlhší počet znakov viac ako jeden film – vtedy by vypísalo len prvý

```
SELECT title, LENGTH(title) AS dlzka FROM film
ORDER BY dlzka DESC
LIMIT 1
```

- **Zistite počet výpožičiek jednotlivých zákazníkov (tab.rental)**
  - výsledok vyžaduje spojenie tabuľky s údajmi o zákazníkoch a o výpožičkách
  - vytvoríme skupiny výpožičiek podľa zákazníkov – ak by sa náhodou dvaja rôzni zákazníci volali rovnako, vystupovali by v skupinách podľa mena a priezviska ako jedna identita, preto pridáme do výrazu aj ID zákazníka
  - pre každú takúto skupinu zistíme počet položiek v tabuľke rental

```
SELECT customer.customer_id, customer.first_name, customer.last_name,
COUNT(rental.rental_id) AS pocet FROM customer
INNER JOIN rental ON (customer.customer_id = rental.customer_id)
GROUP BY customer.customer_id, customer.first_name, customer.last_name
```

- **Koľko nevrátených výpožičiek majú jednotliví zákazníci?**
  - analógia s predchádzajúcou úlohou, zobrazíme však len tie položky, ktoré majú prázdne pole return\_date

```
SELECT customer.customer_id, customer.first_name, customer.last_name, rental.return_date,
COUNT(rental.rental_id) AS pocet FROM rental
INNER JOIN customer ON (customer.customer_id = rental.customer_id)
GROUP BY customer.customer_id, customer.first_name, customer.last_name, rental.return_date
HAVING rental.return_date IS NULL
```

- **Zistite počet výpožičiek v jednotlivých mesiacoch roku 2005 (tab. rental)**

- podobne ako vieme z dátumu získať rok (cez YEAR), vieme získať aj mesiac (MONTH)
- vytvoríme skupiny podľa rokov a mesiacov a zistíme v nich počty
- pridáme podmienku, aby sa uvažoval len rok 2005

```
SELECT MONTH(rental_date) AS mesiac, YEAR(rental_date) AS rok, COUNT(*) AS pocet FROM
rental
GROUP BY rok, mesiac
HAVING rok = 2005
```

- **Pre každú kategóriu filmov uveďte priemernú dĺžku filmov.**

- spojíme tabuľku filmov a kategórií cez film\_category a jednoducho len zobrazíme potrebné stĺpce a zoskupíme podľa kategórií

```
SELECT category.name, AVG(film.length) as priemer FROM film
INNER JOIN film_category ON (film.film_id = film_category.film_id)
INNER JOIN category ON (category.category_id = film_category.category_id)
GROUP BY category.name
```

- **Vytvorte dotaz, ktorý zobrazí názov krajiny, názov mesta a počet zákazníkov, ktorí majú v tomto meste adresu**

- spojíme tabuľky country, city, address a customer cez príslušné ID a do výsledku zahrnieme len príslušnú trojicu stĺpcov

```
SELECT country.country, city.city, COUNT(customer.customer_id) AS pocet FROM country
INNER JOIN city ON (country.country_id = city.country_id)
INNER JOIN address ON (city.city_id = address.city_id)
INNER JOIN customer ON (customer.address_id = address.address_id)
GROUP BY country.country, city.city
ORDER BY pocet DESC
```

- **Pre každého herca uveďte meno a priezvisko a pridajte stĺpec s jeho iniciálkami**

- jednoduchá úloha, vyberieme prvý znak z reťazcov (SUBST) a spojíme ich do jedného (CONCAT)

```
SELECT first_name, last_name, CONCAT(SUBSTR(first_name,1,1),SUBSTR(last_name,1,1)) AS
inicialky FROM actor
```

- **Vypíšte názvy filmov v obrátenom poradí sprava doľava ☺**

- ide len o použitie funkcie, konkrétne REVERSE

```
SELECT REVERSE(title) AS opacne FROM films
```

- **Pre každého zákazníka vytvorte heslo obsahujúce prvé písmeno mena a priezvisko a spravte z neho hash (MD5 a SHA1).**

- prvé písmeno mena a priezvisko spojíme pomocou CONCAT a na výsledok pustíme hashovaciu funkciu

```
SELECT MD5(CONCAT(LEFT(first_name,1), last_name)) AS heslo FROM customer
```

- **Vypíšte názov filmu, meno a priezvisko herca, ktorý v ňom hral**

- jednoduché spojenie cez tri tabuľky

```
SELECT film.title, actor.first_name, actor.last_name FROM actor
INNER JOIN film_actor ON (actor.actor_id = film_actor.actor_id)
INNER JOIN film ON (film_actor.film_id = film.film_id)
ORDER BY film.title
```

- **Spočítajte počet vypožičaných filmov v jednotlivých kategóriách**

- hoci zadanie vyzerá na prvý pohľad zložito, postačí nám napojiť kategórie cez film\_category na tabuľku inventory, v ktorej je zoznam požičiavaných filmov
- k názvu filmu sa nepotrebujeme dostávať, ich počet vieme zistiť aj cez jedinečné film\_id v tabuľke film\_category alebo inventory

```
SELECT category.name, COUNT(DISTINCT(inventory.film_id)) FROM category
INNER JOIN film_category ON (film_category.category_id = category.category_id)
INNER JOIN inventory ON (film_category.film_id = inventory.film_id)
GROUP BY category.name
```