

## Databáze v.3

Test je členěn na oblast databázové teorie a na praktické úkoly v jazyce SQL (použijte DB jazyk, který Vám vyhovuje).

Prosíme, abyste si pozorně přečetl(a) následující instrukce před spuštěním testu:

- Každou odpověď je potřeba uložit stisknutím tlačítka DALŠÍ, které se nachází pod každou odpovědí.
- Ke každé otázce se v průběhu trvání testu můžete kdykoliv vrátit, a svou odpověď jakkoliv pozměnit či upravit a následně ji znovu uložit.
- Test není možné přerušit. V případě, že by došlo k neočekávanému přerušení testu, bude Vám vygenerován nový test po domluvě.
- Test je časově náročný, doporučujeme si vyhradit celých 60 minut pouze pro tento test.

### 1.

#### Vysvětlete, co znamenají následující zkratky a pojmy?

DB TEORIE

RDBMS, PRIMARY KEY, FOREIGN KEY, ODBC, JDBC, XML, OLTP, OLAP, B-Tree, Hadoop, MPP (Massively Parallel Processing)

##### Odpověď:

##### Hodnocení:

<b>RDBMS</b>	relační databázový systém - systém dat v tabulkách, které mezi sebou mají relace a splňují pravidla normalizace	Procentní ohodnocení: 100
<b>PRIMARY KEY</b>	primární klíč - unikátní identifikátor řádku	
<b>FOREIGN KEY</b>	cizí klíč - relace v jedné tabulce na unikátní klíč v jiné tabulce	
<b>ODBC, JDBC</b>	API - rozhraní pro přístup k databázi pomocí programovacího jazyk	
<b>XML</b>	značkovací jazyk - standardizovaný jazyk pro strukturalizaci dat a možnost přenosu v rámci různých platforem	
<b>OLTP, OLAP</b>	transakční databáze - probíhá tam mnoho úprav dat. Narodil od OLAP, který slouží pro ukládání a analýzu dat	
<b>B-Tree</b>	system uložení dat v paměti, databázi...	
<b>Hadoop</b> <b>MPP (Massively Parallel Processing)</b>		

### 2.

#### K čemu je používán E-R model? Co si představujete pod pojmy Entita a Atribut?

DB TEORIE

##### Odpověď:

##### Hodnocení:

Entitně - vztahový model dat.

Procentní ohodnocení: 100

Entita je abstrakce schopná samostatné existence.  
Atribut je vlastnost entity.

### 3.

#### Vysvětlete rozdíl mezi uloženou procedurou a triggerem.

**Odpověď:**

Uložená procedura se spouští na základě jejího volání příkazem EXEC. Procentní ohodnocení: 100  
Trigger se spustí automaticky na základě např. update, insertu či delete do tabulky.

**Hodnocení:****4.**

**Příkazy SQL mohou být rozděleny do několika skupin. Nejznámější jsou DDL (Data Definition Language) a DML (Data Manipulation Language). Uvedte typické příkazy patřící do těchto skupin. Jaké další skupiny znáte?**

**Odpověď:**

DDL: SELECT, INSERT, UPDATE, DELETE

DML: CREATE, DROP, ALTER  
DCL: GRANT, REVOKE COMMIT, ROLLBACK  
EXPLAIN: MERGE

**Hodnocení:**

Procentní ohodnocení: 100

**5.**

**Napište SQL příkaz, který vytvoří tabulku okres tak, aby mohla být použita v DWH.**

**Odpověď:**

CREATE TABLE DimOkres (

OkresID int not null,  
Name varchar(50)  
)

**Hodnocení:**

Procentní ohodnocení: 100

**6.**

**Napište SQL příkaz, který odstraní tabulku Customer.**

**Odpověď:**

DROP TABLE Customer

**Hodnocení:**

Procentní ohodnocení: 100

**7.**

**Napište SQL příkaz, který do tabulky "LogFiles" (do které aplikace ukládá informaci o zpracovaných souborech) přidá povinný číselný sloupec FileSize. Velikost souboru může nabývat až 5GB.**

**Odpověď:**

**Automatické hodnocení (není odpověď):**

0%

**Hodnocení:**

Procentní ohodnocení: 0

Výsledek automatického vyhodnocení (není odpověď)

**8.**

**Co je view a k čemu je používáno - uveďte typické případy využití? Napište příklad vytvoření view v jazyce SQL.**

DB TEORIE

**Odpověď:**

Pohled - jedná se o výpis z jedné nebo více tabulek - samotný neobsahuje žádná data, lze přez něj i upravovat data v tabulce prostřednictvím jiných oprávnění.

```
CREAT VIEW V_Customers AS (SELECT Name, City FROM Customers)
```

**Hodnocení:**

Procentní ohodnocení: 100

**9.**

**Co je datový slovník (data dictionary) v RDBMS? Jaké informace jsou v něm uloženy? Uveďte příklad zjištění nějaké informace pomocí SQL z datového slovníku RDBMS, který znáte.**

DB TEORIE

**Odpověď:****Automatické hodnocení (není odpověď):**

0%

**Hodnocení:**

Procentní ohodnocení: 0

Výsledek automatického vyhodnocení (není odpověď)

**10.**

**Co je to kaskádovitý delete? Vysvětlete jeho princip.**

DB TEORIE

**Odpověď:****Automatické hodnocení (není odpověď):**

0%

**Hodnocení:**

Procentní ohodnocení: 0

Výsledek automatického vyhodnocení (není odpověď)

**11.**

**Co je to doménová a referenční integrita? Kterým příkazem SQL se dají nastavit?**

DB TEORIE

**Odpověď:****Automatické hodnocení (není odpověď):**

0%

**Hodnocení:**

Procentní ohodnocení: 0

Výsledek automatického vyhodnocení (není odpověď)

12.

Tabulka Customer obsahuje sloupec StatusID. V tomto sloupci je vždy pouze aktuální hodnota. Navrhněte úpravu modelu tak, aby byla k dispozici historie všech stavů, které kdy zákazník měl.

DB TEORIE



Odpověď:

Hodnocení:

Procentní ohodnocení: 0

13.

Mějme tabulku PLAYER, v níž budou sloupce: BIB, PlayerName, Team. Sloupec BIB je primárním klíčem tabulky. Popište, co bude výsledkem následujícího příkazu SQL?

```
SELECT a.PlayerName SideA, b.PlayerName SideB
FROM PLAYER a, PLAYER b
WHERE a.BIB<>b.BIB
```

PRAKTICKÉ SQL ÚKOLY

PLAYER		
BIB	numeric	<pk>
PlayerName	varchar(200)	
Team	varchar(200)	

Odpověď:

Hodnocení:

Nekonečný počet záznamů.

Procentní ohodnocení: 0

14.

Mějme tabulku vytvořenou následujícím příkazem:

```
CREATE TABLE Sales (Id int, Id_Customer int, SaleDate datetime);
```

Napište SQL příkaz, který zobrazí dva sloupce - SaleDate a FirstOfYear, kde sloupec FirstOfYear bude zobrazovat 1. ledna stejného roku jako sloupec SaleDate (např. 25.8.2010 => 1.1.2010). Použijte DB jazyk, který vám vyhovuje.

PRAKTICKÉ SQL ÚKOLY

Odpověď:

Hodnocení:

```
SELECT SaleDate, ( '1.1.'+CONVERT(varchar,YEAR(SaleDate))) AS
FirstOfYear FROM Sales
```

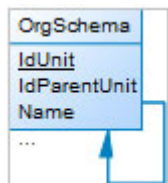
Procentní ohodnocení: 100

15.

Databáze obsahuje tabulku OrgSchema (viz model). Předpokládejme, že naše

**organizační struktura má dvě až tři úrovně: holding, podnik a nepovinně pobočka. Napište příkaz SQL, kterým získáte tabulku se sloupci HOLDING, PODNIK, POBOČKA. Ve sloupcích budou názvy jednotlivých společností.**

PRAKTICKÉ SQL ÚKOLY



**Odpověď:**

```
CREATE TABLE ParentUnit (
    IdParentUnit int NOT NULL,
    HOLDING varchar(50) NOT NULL,
    PODNIK varchar(50) NOT NULL,
    [POBOČKA] varchar(50) NULL
)
```

**Hodnocení:**

Procentní ohodnocení: 0

**16.**

**V databázi je vytvořena uživatelská funkce fn\_NormalizeNumber, která konvertuje telefonní číslo do mezinárodního formátu. Testováním bylo zjištěno, že funkce nepracuje správně. Navrhněte opravu funkce.**

PRAKTICKÉ SQL ÚKOLY

Vyberte si databázový jazyk, který vám vyhovuje. Skripty jsou obsahově identické.

**MS SQL Server**

```
1 CREATE FUNCTION fn_NormalizeNumber(@num varchar(20)) RETURNS varchar(20)
2 BEGIN
3     DECLARE @number varchar(20);
4     IF @num LIKE '+%'
5         SET @number=@num;
6     ELSE IF @num LIKE '00%'
7         SET @number='+' + SUBSTRING(@num,3,20)
8     ELSE
9         SET @number='+420'+@num
10    RETURN @number
11 END
```

**Oracle**

```
1 CREATE OR REPLACE FUNCTION fn_NormalizeNumber(v_num varchar2(20))
2     RETURN varchar2(20) DETERMINISTIC
3 IS v_number varchar2(20);
4 BEGIN
5     IF v_num LIKE '+%' THEN
6         v_number := v_num;
7     ELSE IF v_num LIKE '00%' THEN
8         v_number := '+' || substr(v_num, 3);
9     ELSE
10        v_number := '+420' || v_num;
11    END IF
12    RETURN v_number;
13 END;
```

**Odpověď:**

RETURN @number

**Hodnocení:**

Procentní ohodnocení: 100

**17.**

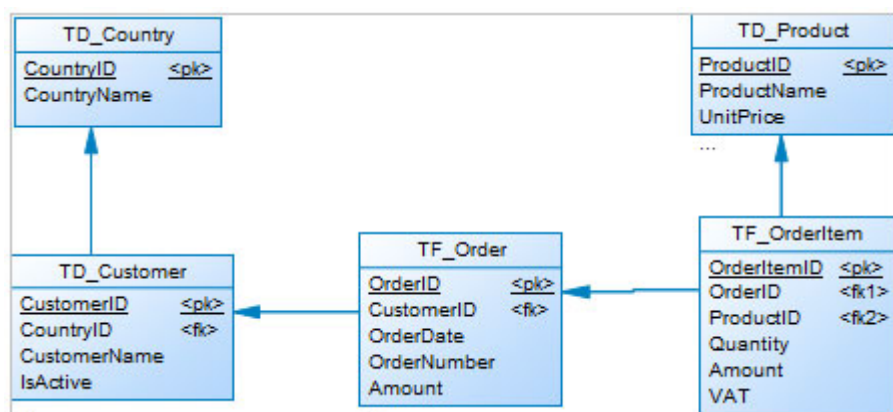
**Databáze objednávek je vytvořena na základě uvedeného modelu. Napište SQL příkaz, který vypíše:**

**Zákazníky, kteří v minulém roce objednali zboží v celkové sumě alespoň za 10.000 Kč.**

**Pro každého uveďte jméno a celkovou sumu jimi odebraného zboží za rok.**

**Zákazník s největším obrátem by měl být uveden na prvním řádku. Výsledek omezte na max 20 řádků.**

PRAKTICKÉ SQL ÚKOLY

**Odpověď:**

```
SELECT TOP 20 CustomerName, SUM(p.UnitPrice*oi.Quantity)
```

```
FROM TD_Customer c
LEFT JOIN TF_Order o ON o.CustomerID=c.CustomerID
LEFT JOIN TF_OrderItem oi ON oi.OrderID=o.OrderID
LEFT JOIN TD_Product p ON p.ProductID=oi.ProductID
ORDER BY SUM(p.UnitPrice*oi.Quantity) DESC
```

**Hodnocení:**

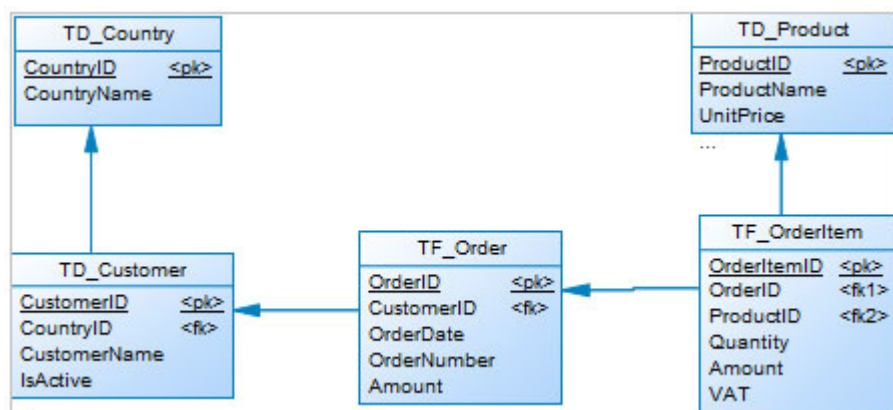
Procentní ohodnocení: 90  
chybí having

**18.**

**Databáze objednávek je vytvořena na základě uvedeného modelu.**

**Napište příkaz SQL, který ODSTRANÍ všechny neaktivní zákazníky.**

PRAKTICKÉ SQL ÚKOLY



**Odpověď:**

**Hodnocení:**

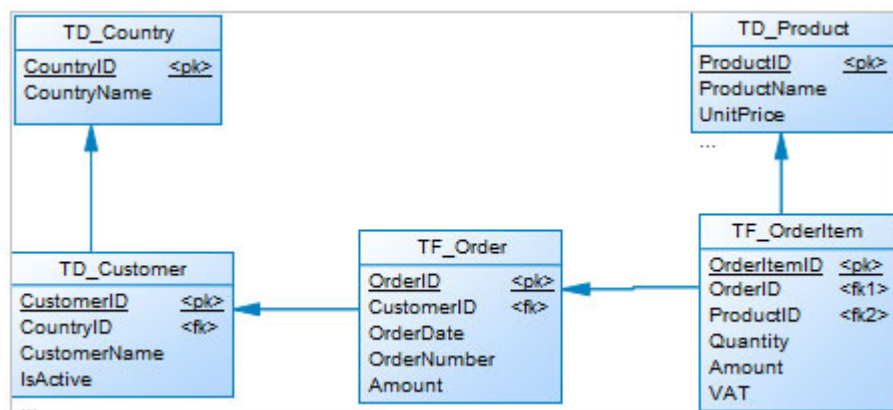
Procentní ohodnocení: 0

**19.**

**Databáze objednávek je vytvořena na základě uvedeného modelu.**

**Napište příkaz SQL, který o 10% SNÍŽÍ cenu všech produktů, které NIKDY NEBYLY OBJEDNÁNY.**

PRAKTICKÉ SQL ÚKOLY



**Odpověď:**

**Hodnocení:**

UPDATE p SET p.UnitPrice=p.UnitPrice\*0.9

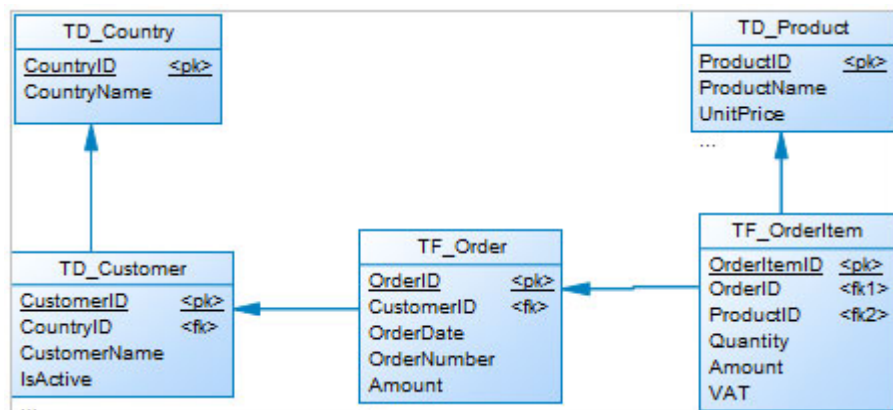
Procentní ohodnocení: 75

Product p INNER JOIN TF\_OrderItem oi ON oi.ProductID=p.ProductID  
WHERE oi.OrderItemID IS NULL

**20.**

**Databáze objednávek je vytvořena na základě uvedeného modelu. Pracovní tabulka tmpCustomer má stejnou strukturu jako tabulka TD\_Customer a obsahuje nově napočtené záznamy zákazníků. Hodnoty primárních klíčů pro stejné zákazníky jsou v obou tabulkách shodné. Napište příkaz, který aktualizuje jména zákazníků v tabulce TD\_Customer a vloží do ní nové zákazníky.**

PRAKTICKÉ SQL ÚKOLY



**Odpověď:**

**Hodnocení:**

Procentní ohodnocení: 0

21.

Tabulka zaměstnanců je vytvořena pomocí uvedeného scriptu. Tabulka obsahuje cca 30.000 záznamů. Dotaz uvedený na řádku 4 je přesto neúměrně pomalý. Upravte ho tak, aby pracoval rychleji. Vysvětlete důvod úpravy.

PRAKTICKÉ SQL ÚKOLY

```
1 CREATE TABLE Employee (EmpCode char(8), EmpName varchar(200))
2 CREATE INDEX iEmpl ON Employee(EmpName)
3
4 SELECT * FROM Employee WHERE SUBSTRING(EmpName, 1, 1)='X'
```

Odpověď:

Hodnocení:

Procentní ohodnocení: 0

22.

Tabulka studentů byla vytvořena následujícím příkazem:

```
CREATE TABLE Student (
  StudentID int PRIMARY KEY,
  Name varchar2(200),
  SocialNo varchar2(10)
);
```

Jak zajistíte, aby se v tabulce nemohl stejný student objevit vícekrát?

PRAKTICKÉ SQL ÚKOLY

Odpověď:

Hodnocení:

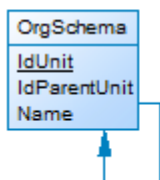
SocialNo varchar2(10) NOT NULL UNIQUE

Procentní ohodnocení: 100

23.

Tabulka OrgSchema obsahuje jednotlivé organizační jednotky v podniku. Napište SQL dotaz, který zobrazí jednotlivé úrovně řízení pro každou organizační jednotku. Výpis obsahuje příklad očekávaného výstupu.

PRAKTICKÉ SQL ÚKOLY



Level	Name
1	director
2	HR management
2	Development management
3	HR recruitment
3	Application Development team
3	Report development team

Odpověď:

**Automatické hodnocení (není odpověď):**

0%



**Hodnocení:**

Procentní ohodnocení: 0

Výsledek automatického vyhodnocení (není odpověď)