

1_labor

1. Kérdezzük le azon dolgozók vezetéknevét és felvételük dátumát, akiket 2002. február 1. és 2006. július 31. között vettek fel!
2. Kérdezzük le, hogy az egyes dolgozók mely országban dolgoznak! A dolgozók teljes neve szerepeljen az eredményben!
3. Kérdezzük le, hogy az egyes munkakörökhöz mekkora átlagos bér tartozik! A munkakörök azonosítójukkal és nevükkel legyenek azonosítva! Csak azok a munkakörök szerepeljenek az eredményben, ahol az átlagbér legalább \$8000!
4. Kérdezzük le, hogy melyek azok a részlegek, ahol több mint 5 ember dolgozik! A részlegek nevükkel és azonosítójukkal legyenek azonosítva!
5. Kérdezzük le azon munkakörök nevét, amelyre 2003-ban vettek fel alkalmazottat!
6. Kérdezzük le, hogy összesen hányan dolgoznak azokon a részlegeken, ahol az átlagos fizetés több, mint \$10000!
7. Kérdezzük le, hogy mekkora az egyes dolgozók fizetésének eltérése az azonos munkakörben dolgozók átlagától? A dolgozók azonosítójukkal szerepeljenek az eredményben!
8. Kérdezzük le azon dolgozók teljes nevét, azonosítóját és beosztásuk azonosítóját, akik olyan részlegen dolgoznak, amelynek helyszíne Seattle-ben van!

2_labor

1. Írjon programot, amely eldönti egy számról, hogy az páros vagy páratlan!
2. Írjon programot, amely megadja, hogy három szám közül melyik a legnagyobb!

3_labor

1. Írjon programot, amely megadja, hogy egy adott azonosítójú dolgozónak ki a menedzsere és hogy az melyik részleg dolgozója!
2. Írjon programot, amely eldönti, hogy a 160-as és 162-es azonosítójú dolgozók fizetése közül melyik a nagyobb, majd a kisebb fizetést a nagyobb értékre állítja!
3. Írjon programot, amely meghatározza, hogy egy adott azonosítójú dolgozónak hány év tapasztalata van, majd a következőknek megfelelően növeli a fizetését: tapasztalat > 15: +25%; 15 >= tapasztalat > 10: +15%; 10 >= tapasztalat > 5: +5%; egyébként: +2%.
4. Írjon programot, amely ellenőrzi, hogy a dolgozó azonosítók folytonosak-e vagy sem (azaz vannak-e hiányzó azonosítók)!
5. Írjon programot, amely havi bontásban megjeleníti, hogy hány dolgozót vettek fel egy adott év (pl. 2005) egyes hónapjaiban!

4_labor

1. Írjon programot, amely eltárolja egy adott keresztnévű dolgozó adatait egy rekordban, majd kiírja a teljes nevét és a felvételének dátumát (formátum: YYYY-MM-DD)! Kezelje a lehetséges kivételeket!
2. Írjon programot, amely generál 10 véletlen dolgozó azonosítót (10 egész számot 100 és 206 között), majd ezeket és az ezekhez tartozó fizetés értékeket eltárolja egy dinamikus tömbben, végül pedig kiírja a tárolt adatokat!
3. Írjon olyan programot, amely meghatározza, hogy melyek azok a dolgozók, akik a beosztásukhoz tartozó maximális fizetés legalább 75%-át vagy többet kapnak! A dolgozók azonosítóját és fizetésük százalékos eltérést egy dinamikus tömbben tárolja, majd végül jelenítse meg!

5_labor

1. Írjon programot, amely megadja, hogy a vállalat USA-beli telephelyei melyik állam melyik városában vannak! A megvalósításhoz használjon beágyazott táblát!
2. Hozzon létre egy EMPLOYEES_SEN nevű táblát, amelynek szerkezete megegyezik az EMPLOYEES tábla szerkezetével! Ezt követően írjon programot, amely átmásolja ebbe a táblába azon dolgozók adatait, akik az egyes részlegeken a legrégebben dolgoznak, úgy, hogy a fizetésüket 25%-kal növeli! A megvalósításához használjon asszociatív tömböt!
3. Írjon programot, amely egy asszociatív tömbbe gyűjti a dolgozók azonosítóját, fizetését és felvételük időpontját, majd megadja annak a dolgozónak a teljes nevét és a részleg nevét, ahol dolgozik, aki esetében a legnagyobb a fizetés/tapasztalat hányados!

6_labor

1. Írjon programot, amely kiírja azon dolgozók nevét és fizetését, akik fizetése magasabb, mint egy adott érték! A megvalósításhoz használjon paraméteres kurzort!
2. Módosítsa az előző programot úgy, hogy megadja, hogy az egyes fizetésértékekhez hány dolgozó tartozik!
3. Hozzon létre egy táblát, amely alkalmas a dolgozók vezeték- és keresztnevének, valamint fizetésének tárolására. Majd írjon programot, amely egy előre megadott mennyiségű sorral feltölti ezt a táblát az EMPLOYEES táblából véletlenszerűen kiválasztott adatokat használva!
4. Határozza meg, hogy az egyes részlegek dolgozói a munkaköreikhez tartozó maximális lehetséges fizetés hány százalékát kapják meg átlagosan (azaz egy részleghez egy átlagos százalékértéket kell meghatározni). Jelenítse meg annak a részlegnek a nevét és az ahhoz tartozó százalékértéket (két tizedes jegyre kerekítve), ahol ez az érték a legalacsonyabb! A megvalósításhoz használjon explicit kurzort! (Várt eredmény: Purchasing - 54,34)

7_labor

1. Írjon eljárást, amely egy paraméterként kapott azonosítóhoz tartozó dolgozó fizetését a következő módon növeli az években vett munkatapasztalatának megfelelően:

tapasztalat > 15: +25%; 15 >= tapasztalat > 10: +15%; 10 >= tapasztalat > 5: +5%;
egyébként: +2%.

Az eljárás működtetéséhez írjon programot, amely eltárolja egy beágyazott táblában az 50-es azonosítójú részleg dolgozó azonosítóit és meghívja azokra az eljárást!

Kezelje a lehetséges kivételeket!

2. Írjon függvényt, amely visszatérési értéként megadja, hogy egy paraméterként kapott dolgozó azonosítóhoz hány korábbi beosztás tartozik! A függvény működtetéséhez írjon programot, amely megjeleníti azon dolgozók nevét és a korábbi beosztásaik számát, akiknek volt korábbi beosztása!
3. Hozzon létre a fenti függvényből egy tárolt függvényt, majd írjon egy SQL lekérdezést, amely megjeleníti a dolgozók teljes nevét és a korábbi beosztásaik számát!
4. Írjon tárolt függvényt, amely két dolgozó azonosítót kap paraméterül és visszatérési értéként megadja, hogy melyik dolgozó azonosítóhoz tartozik magasabb fizetés! Ha a két fizetésérték megegyezik, a visszatérési érték legyen 0! Ha pedig nem létezik adott azonosítójú dolgozó, akkor a visszatérési érték legyen -1!

A tárolt függvény működtetéséhez írjon programot, amely meghívja azt a 150 és a 100 paraméterekkel!

8_labor

1. Írjon trigger-t, amely megakadályozza, hogy a dolgozók fizetése csökkenjen! A megvalósításhoz használja a RAISE_APPLICATION_ERROR eljárást!
2. Hozzon létre egy test_table nevű táblát, amelynek egy val_num nevű és NUMBER típusú oszlopa van!

Majd írjon egy trigger-t, amely a pufferbe írja, hogy milyen műveletek (INSERT, UPDATE, DELETE) kerültek végrehajtásra és mikor!

DML utasításokkal ellenőrizze, hogy megfelelően működik-e a trigger!

3. Írjon trigger-t, amely az EMPLOYEES tábla SALARY oszlopának frissítése esetén kiírja a pufferbe, hogy mi volt a dolgozók előző és új fizetése, és hogy mennyi a kettő közötti különbség!

Például:

100 - old: 19200, new: 23040, diff: 3840

101 - old: 13600, new: 16320, diff: 2720

...

4. Hozzon létre egy HR_LOG nevű táblát, amelynek oszlopai a következők:

- MOD_DATE: DATE típusú,

- MOD_ACTION: VARCHAR2(50) típusú,

- EMP_ID: NUMBER(6,0) típusú!

Írjon trigger-t, amely ebbe a táblába naplózza, hogy mikor és hogyan módosították az EMPLOYEES táblát!

9_labor

1. Írjon lokális függvényt, amely paraméterként egy pozitív egész számot kap, visszatérési értéke pedig egy beágyazott tábla, amely azon dolgozók azonosítóit tartalmazza, akik teljes neve rövidebb, mint a paraméterként kapott érték!

Írjon programot, amely a dolgozók ABC sorrendbe rendezett teljes nevei mellett megjeleníti, hogy hány olyan dolgozó van, akinek a teljes neve rövidebb (a fenti függvény használatával)!

A várt eredmény:

Adam Fripp: 8

Alana Walsh: 15

Alberto Errazuriz: 101

...

2. Hozzon létre egy CITY_EMPS nevű táblát! Ennek oszlopai:

- CITY: VARCHAR2(30) típusú, elsődleges kulcs

- NUM_EMP: NUMBER(3) típusú, nem NULL értékű

Írjon lokális eljárást, amely paraméterként egy város nevét kapja, és növeli a CITY_EMPS táblában a városhoz tartozó dolgozók számát 1-gyel, ha a város már szerepel a táblában, egyébként pedig beszúr egy új sort a táblába a város nevével és az 1 értékkel!

Írjon programot, amely feltölti a CITY_EMPS táblát az adatbázis állapotának megfelelően az előzőleg létrehozott eljárás használatával!

3. Írjon lokális eljárást, amely megjeleníti egy adott azonosítójú dolgozó nevét, a korábbi munkaköreinek számát, azok betöltésének kezdő és végdátumát, valamint a munkakörök nevét! Kezelje a lehetséges kivételeket!

Írjon programot, amely megjeleníti a 101-es és a 176-os azonosítójú dolgozók korábbi munkaköreire vonatkozó adatokat a fenti eljárás használatával!

A várt eredmény:

Neena Kochhar had 2 previous jobs:

1997-09-21 - 2001-10-27: Public Accountant

2001-10-28 - 2005-03-15: Accounting Manager

...

4. Írjon lokális függvényt, amely paraméterként kap egy részleg azonosítót, a visszatérési értéke pedig egy beágyazott tábla, amely a részlegben dolgozók munkakör azonosítóit tartalmazza (minden azonosító csak egyszer szerepeljen)!

Írjon programot, amely megjeleníti azon részlegek nevét és a részlegben betöltött különböző munkakörök számát, ahol ez az érték legalább 2 (a fenti függvény segítségével)!

A várt eredmény:

Accounting (2)

Executive (2) ...

10_labor

1. Írjon tárolt függvényt, amely egy dolgozó azonosítót kap paraméterként, visszatérési értéke pedig egy sorszám, amely megadja, hogy az ugyan abban a munkakörben dolgozók fizetései között hányadik az adott dolgozó fizetése csökkenő sorrendet tekintve!

Írjon SQL lekérdezést, amely megadja a 154-es azonosítójú dolgozó teljes nevét és hogy a fizetése hányadik az azonos munkakörben dolgozók esetében (a fenti függvény segítségével)!

A várt eredmény:

Nanette Cambrault 12

2. Írjon tárolt függvényt, amely egy részleg azonosítót kap paraméterként, visszatérési értéke pedig a leggyakoribb munkakör megnevezése az adott részlegen! Ha a részleghez nem szerepel dolgozó az adatbázisban, akkor az "N/A" szöveget adja vissza!

Írjon SQL lekérdezést, amely minden részleg neve mellett megjeleníti a leggyakoribb munkakör nevét!

A várt eredmény:

Accounting Public Accountant

Administration Administration Assistant

Benefits N/A

...

3. Írjon lokális függvényt, amely egy menedzser azonosító kap paraméterként, visszatérési értéke pedig egy olyan rekord, amely az adott menedzserhez tartozó dolgozók átlagos fizetésértékét és a dolgozók számát tartalmazza!

Írjon programot, amely megjeleníti a menedzserek teljes nevét, valamint a hozzájuk tartozó dolgozók számát és az átlagos fizetésértékét kerekítve, ha az utóbbi nagyobb, mint 8000!

A várt eredmény:

Steven King (14): 11100 ...

4. Hozzon létre egy DEPT_STATS nevű táblát! Ennek oszlopai:

- DEPT_ID: NUMBER(4) típusú, elsődleges kulcs

- NUM_EMP: NUMBER(3) típusú

- AVG_EXP: NUMBER(3,1) típusú

Írjon lokális eljárást, amely paraméterként egy részleg azonosítót kap, és beszúr egy új sort a részleg azonosítójával, az adott részlegen dolgozók számával és a dolgozók átlagos munkatapasztalatával (években, egy tizedes pontossággal) a DEPT_STATS táblába, ha a részleg még nem szerepel a táblában!

Írjon programot, amely feltölti a DEPT_STATS táblát azon részlegek adataival, amelyekhez van megadva dolgozó az adatbázisban!