SELECT 1.01

Identyfikatory studentów wraz z datami przystąpienia do egzaminu. Jeśli student danego dnia przystąpił do wielu egzaminów, jego identyfikator ma się pojawić tylko raz. Dane posortowane malejąco względem dat.

**9 rekordów, ostatni: id\_studenta 3, data 2017-02-10**

SELECT DISTINCT id\_studenta, data\_egzaminu

FROM oceny\_studentow

ORDER BY data\_egzaminu DESC

SELECT 1.02

Identyfikatory studentów, którzy przystąpili do egzaminu w marcu 2017 roku. Identyfikator każdego studenta ma się pojawić tylko raz. Dane posortowane malejąco według identyfikatorów studentów.

**2 rekordy, identyfikatory studentów 2 i 3**

SELECT DISTINCT id\_studenta

FROM oceny\_studentow

WHERE data\_egzaminu BETWEEN '20170301' AND '20170331'

ORDER BY id\_studenta DESC

SELECT 1.03

Nazwiska, imiona, daty urodzenia i numery grup studentów nie należących do grupy, która w numerze ma tekst ‘01’. Dane posortowane rosnąco według nazwiska i imienia.

**5 rekordów, pierwszy: Ireneusz Babicz, ostatni: Kamil Nowakowski**

SELECT nazwisko, imie, data\_urodzenia, nr\_grupy

FROM studenci

WHERE nr\_grupy NOT LIKE '%01%'

ORDER BY nazwisko, imie

SELECT 1.04

Wszystkie dane o tych pracownikach, dla których brakuje numeru PESEL lub NIP. Dane posortowane rosnąco według nazwiska a następnie malejąco według imienia.

**6 rekordów, w 4 przypadkach NIP jest NULL, w trzech PESEL jest NULL**

SELECT \*   
FROM pracownicy

WHERE nip IS NULL OR pesel IS NULL

ORDER BY nazwisko, imie DESC

SELECT 1.05

Liczbę wszystkich wykładowców. Kolumna ma mieć nazwę Liczba\_wykladowcow.

**W bazie danych jest sześciu wykładowców (sprawdź w tabeli Wykładowcy)**

SELECT COUNT(id\_wykladowcy) AS liczba\_wykladowcow

FROM wykladowcy

SELECT 1.06

Sumę godzin wszystkich wykładów. Kolumna ma mieć nazwę suma\_godzin.

**Zapytanie zwraca liczbę 435**

SELECT SUM(liczba\_godzin) AS suma\_godzin

FROM wyklady

SELECT 1.07

Imiona, nazwiska i daty urodzenia studentów, którzy urodzili się w lutym. Dane posortowane malejąco według daty urodzenia.

Wskazówka:

W zapytaniu należy użyć funkcji DatePart. Zwraca ona liczbę całkowitą oznaczającą wybraną część daty (rok, miesiąc lub dzień). Funkcja wymaga podania dwóch parametrów. Jako pierwszy podajemy DAY, MONTH lub YEAR wskazując, która część daty ma zostać zwrócona. Jako drugi parametr podajemy datę, z której wskazana część ma zostać wybrana.

Przykład:

DatePart(Day, ‘20180221’) zwróci liczbę 21  
DatePart(Month, ‘20190411’) zwróci liczbę 4

**3 rekordy. Pierwszy: Wioletta Lubomirska ur. 20-02-1990**

SELECT nazwisko, imie, data\_urodzenia

FROM studenci

WHERE datepart(month, data\_urodzenia)=2

ORDER BY data\_urodzenia DESC

SELECT 1.08

Identyfikatory studentów wraz z numerami grup, zapisanych do grup o nazwach zaczynających się na DMIe oraz identyfikatory wykładowców wraz z nazwami katedr, w których pracują. Pierwsza kolumna ma mieć nazwę student\_wykladowca a druga grupa\_katedra. Dane posortowane rosnąco według ostatniej kolumny.

**11 rekordów, 5 studentów i 6 wykładowców,**

**ostatni rekord: identyfikator:1, Katedra Matematyki**

SELECT id\_studenta AS student\_wykladowca, nr\_grupy AS Grupa\_Katedra

FROM studenci WHERE nr\_grupy LIKE 'DMIe%'

**UNION**

SELECT id\_wykladowcy, katedra

FROM wykladowcy

ORDER BY grupa\_katedra

SELECT 1.09

Liczbę wykładów, które gdziekolwiek w nazwie mają literę m. Kolumna ma mieć nazwę ZLiteraM.

**Jest 9 takich wykładów.**

SELECT COUNT(id\_wykladu) AS ZLiteraM

FROM wyklady

WHERE nazwa\_wykladu LIKE '%m%'

SELECT 1.10

Pary id\_studenta, id\_wykladu z tabeli studenci\_wyklady, którym nie została przyznana dotychczas żadna ocena. Wykorzystaj operator EXCEPT.

**Takich par jest 8.**

SELECT id\_studenta, id\_wykladu  
FROM studenci\_wyklady

**EXCEPT**

SELECT id\_studenta, id\_wykladu  
FROM oceny\_studentow

SELECT 1.11

Identyfikatory studentów, którzy zapisali się zarówno na wykład o identyfikatorze 4 jak i 11.  
Wykorzystaj operator INTERSECT.

**Jest tylko jeden taki student: o identyfikatorze 6**

SELECT id\_studenta

FROM studenci\_wyklady

WHERE id\_wykladu=4

**INTERSECT**

SELECT id\_studenta

FROM studenci\_wyklady

WHERE id\_wykladu=11

SELECT 1.12a

Identyfikatory wszystkich doktorów.

**Jest trzech doktorów, o identyfikatorach 1, 3 i 9**

SELECT id\_wykladowcy

FROM wykladowcy

WHERE stopien\_tytul='doktor'

SELECT 1.12b

Identyfikatory wszystkich doktorów, którzy prowadzą co najmniej jeden wykład.

**Co najmniej jeden wykład prowadzą doktorzy o identyfikatorach 1 i 3**

SELECT id\_wykladowcy

FROM wykladowcy

WHERE stopien\_tytul='doktor'

INTERSECT

SELECT id\_wykladowcy

FROM wyklady

SELECT 1.12c

Identyfikatory wszystkich doktorów, którzy nie prowadzą żadnego wykładu.

**Wykładu nie prowadzi doktor o identyfikatorze 9**

SELECT id\_wykladowcy

FROM wykladowcy

WHERE stopien\_tytul='doktor'

EXCEPT

SELECT id\_wykladowcy

FROM wyklady

SELECT 1.13

Liczbę wykładów (pod nazwą liczba\_wykladow) oraz łączną sumę godzin (pod nazwą liczba\_godzin) przewidzianych na wykłady, których nazwy zaczynają się na In.

**Takich wykładów jest 5, suma godzin jest równa 165**

SELECT COUNT(id\_wykladu) liczba\_wykladow, SUM(liczba\_godzin) liczba\_godzin

FROM wyklady

WHERE nazwa\_wykladu LIKE 'In%'

SELECT 1.14

Liczbę pracowników, dla których brakuje numeru PESEL i NIP. Jedyna zwracana kolumna ma mieć nazwę brak\_PESELiNIP

**Zapytanie zwraca liczbę 1**

SELECT count(\*) AS brak\_PESELiNIP

FROM pracownicy

WHERE PESEL IS NULL AND NIP IS NULL

SELECT 1.15

Datę ostatniego egzaminu (użyj funkcji max). Zwracaną kolumnę nazwij ostatniEgzamin

**Ostatni egzamin odbył się 13.10.2017**

SELECT max(data\_egzaminu) AS ostatniEgzamin

FROM oceny\_studentow

SELECT 1.16

Liczbę doktorów pracujących w Katedrze Informatyki

**W Katedrze Informatyki pracuje jeden doktor**

SELECT count(\*) liczbaDoktorow

FROM wykladowcy

WHERE katedra='Katedra Informatyki' AND stopien\_tytul='doktor'