SELECT 3.01

Nazwy grup, do których zapisany jest przynajmniej jeden student wraz z liczbą tych studentów. Kolumna zwracająca liczbę studentów ma mieć nazwę Liczba\_studentow. Dane posortowane rosnąco według liczby studentów.

**Do trzech grup zapisani sią studenci, do dwóch po pięciu i do jednej dwóch**

SELECT nr\_grupy, COUNT(\*) AS Liczba\_studentow

FROM studenci

GROUP BY nr\_grupy

ORDER BY Liczba\_studentow

SELECT 3.02

Nazwy grup, do których zapisanych jest przynajmniej trzech studentów wraz z liczbą tych studentów. Kolumna zwracająca liczbę studentów ma mieć nazwę Liczba\_studentow. Dane posortowane rosnąco według liczby studentów.

**Takie grupy są dwie**

SELECT nr\_grupy, COUNT(\*) AS Liczba\_studentow

FROM studenci

GROUP BY nr\_grupy

HAVING COUNT(\*)>=3

ORDER BY Liczba\_studentow

SELECT 3.03

Identyfikatory, imiona, nazwiska i numery grup studentów, którzy nie są zapisani na żaden wykład. Dane posortowane malejąco według nazwisk i imion.

**4 rekordy. Ostatni: Klaudia Nowak**

SELECT id\_studenta, imie, nazwisko, nr\_grupy

FROM studenci

WHERE id\_studenta NOT IN (SELECT id\_studenta FROM studenci\_wyklady)

ORDER BY nazwisko, imie

SELECT 3.04

Imiona, nazwiska i numery grup studentów z grup, których nazwa zaczyna się na DMIe i którzy nie są zapisani na wykład z Informatyki

**5 rekordów**

SELECT imie, nazwisko, nr\_grupy

FROM studenci

WHERE nr\_grupy LIKE 'DMIe%' AND id\_studenta NOT IN

(SELECT id\_studenta FROM studenci\_wyklady sw INNER JOIN wyklady w ON sw.id\_wykladu=w.id\_wykladu

WHERE nazwa\_wykladu='Informatyka')

SELECT 3.05

Nazwiska i imiona studentów o nazwisku Nowakowski, którzy otrzymali przynajmniej jedną ocenę wraz ze średnią ocen (każdego Nowakowskiego z osobna). Kolumna zwracająca średnią ma mieć nazwę Srednia. Dane posortowane malejąco według nazwisk i malejąco według imion.

**Nowakowski Kamil, średnia 4.75**

SELECT nazwisko, imie, AVG(ocena) AS Srednia

FROM studenci s INNER JOIN oceny\_studentow os ON s.id\_studenta=os.id\_studenta

WHERE nazwisko='Nowakowski'

GROUP BY s.id\_studenta, nazwisko, imie

ORDER BY nazwisko DESC, imie DESC

SELECT 3.06

Nazwiska i imiona wykładowców, którzy prowadzą co najmniej jeden wykład wraz ze średnią ocen jakie dali studentom (jeśli wykładowca nie dał do tej pory żadnej oceny, także ma się pojawić na liście). Kolumna zwracająca średnią ma mieć nazwę Srednia. Dane posortowane malejąco według średniej.

**4 rekordy. Pierwszy rekord: Jacek Kwiatkowski, średnia 4.50**

**Jeden wykładowca nie wystawił oceny**

SELECT nazwisko, imie, AVG(ocena) AS srednia

FROM pracownicy INNER JOIN wyklady ON id\_pracownika= id\_wykladowcy

LEFT JOIN oceny\_studentow ON wyklady.id\_wykladu=oceny\_studentow.id\_wykladu

GROUP BY wykladowcy.id\_wykladowcy, nazwisko, imie

ORDER BY srednia DESC

SELECT 3.07

Identyfikatory, nazwiska i imiona studentów zapisanych tylko na jeden wykład. Dane posortowane rosnąco według identyfikatorów studentów.

**4 rekordy. Pierwszy rekord: Ireneusz Babicz, id=7**

SELECT s.id\_studenta, nazwisko, imie

FROM studenci s INNER JOIN studenci\_wyklady sw ON s.id\_studenta=sw.id\_studenta

GROUP BY s.id\_studenta, nazwisko, imie

HAVING COUNT(s.id\_studenta)=1

ORDER BY s.id\_studenta

SELECT 3.08

Wszystkie możliwe oceny oraz ile razy każda z ocen została przyznana (kolumna ma mieć nazwę Liczba\_ocen). Dane posortowane malejąco według ocen.

**6 rekordów. Ocena 2 i 5 zostały przyznane po 3 razy, każda z pozostałych jeden raz**

SELECT o.ocena, COUNT(o.ocena) AS Liczba\_ocen

FROM oceny o LEFT JOIN oceny\_studentow os ON o.ocena=os.ocena

GROUP BY o.ocena

ORDER BY o.ocena

SELECT 3.09

Nazwy wykładów oraz kwotę, jaką uczelnia musi przygotować na wypłaty pracownikom prowadzącym wykłady ze Statystyki i Informatyki (osobno). Jeśli jest wiele wykładów ze Statystyki lub Informatyki, suma dla nich ma być obliczona łącznie. Zapytanie ma więc zwrócić dwa rekordy (jeden dla wykładów ze Statystyki, drugi dla Informatyki).

Kwotę za jeden wykład należy obliczyć jako iloczyn stawki godzinowej danego pracownika oraz liczby godzin przeznaczonych na prowadzone przez niego wykłady.

**Informatyka 7200, Statystyka 675**

SELECT nazwa\_wykladu, SUM(stawka\*liczba\_godzin)

FROM (stopnie\_tytuly st INNER JOIN wykladowcy w ON st.stopien\_tytul=w.stopien\_tytul)

INNER JOIN wyklady wk ON wk.id\_wykladowcy=w.id\_wykladowcy

WHERE nazwa\_wykladu in ('Statystyka','Informatyka')

GROUP BY nazwa\_wykladu

SELECT 3.10

Nazwiska i imiona wszystkich wykładowców wraz z sumaryczną liczbą godzin prowadzonych przez nich wykładów. Kolumna zwracająca liczbę godzin ma mieć nazwę Liczba\_godzin. Dane posortowane malejąco według liczby godzin.

**6 rekordów. Pierwszy Jacek Kwiatkowski 195 godzin.**

**Ostatnie dwa rekordy mają liczbę godzin NULL.**

SELECT nazwisko, imie, SUM(liczba\_godzin) AS Liczba\_godzin

FROM (pracownicy INNER JOIN wykladowcy ON id\_pracownika=id\_wykladowcy)

LEFT JOIN wyklady w ON w.id\_wykladowcy=wykladowcy.id\_wykladowcy

GROUP BY nazwisko, imie, w.id\_wykladowcy

ORDER BY Liczba\_godzin DESC

SELECT 3.11

Nazwy wykładów z Historii wraz z datami egzaminów. Jeśli tego samego dnia było wiele egzaminów z danego wykładu, tabela wynikowa zapytania ma zwrócić tylko jeden rekord dotyczący tego wykładu. Dane posortowane malejąco według daty.

UWAGA: należy uwzględnić fakt, że w bazie danych może być wiele różnych wykładów z Historii (mających różne identyfikatory). W takiej sytuacji mają one być traktowane jako osobne wykłady.

**3 rekordy. W pierwszym jest data 11-10-2017**

SELECT nazwa\_wykladu, data\_egzaminu

FROM wyklady w INNER JOIN oceny\_studentow os ON w.id\_wykladu=os.id\_wykladu

WHERE nazwa\_wykladu='Historia'

GROUP BY w.id\_wykladu, nazwa\_wykladu, data\_egzaminu

ORDER BY data\_egzaminu DESC

SELECT 3.12

Nazwy wszystkich grup oraz liczbę studentów należących do każdej grupy (kolumna ma mieć nazwę LiczbaStudentow). Dane posortowane rosnąco według liczby studentów.

**4 rekordy. Do grupy ZMZm2001 nie jest zapisany żaden student**

SELECT g.nr\_grupy, COUNT(id\_studenta) AS LiczbAStudentow

FROM grupy g LEFT JOIN studenci s ON g.nr\_grupy=s.nr\_grupy

GROUP BY g.nr\_grupy

ORDER BY LiczbaStudentow

SELECT 3.13

Nazwy wykładów, które na początku mają słowo Historia wraz z liczbą studentów zapisanych na te wykłady (kolumna ta ma mieć nazwę Liczba\_studentow).

UWAGA: jeśli uruchomiono wiele wykładów o tej samej nazwie, każdy ma się pojawić osobno. Zapytanie ma zwracać także dane o wykładach, na które nie jest zapisany żaden student. Dane posortowane malejąco według liczby zapisanych studentów.

**3 rekordy. Liczba zapisanych studentów wynosi kolejno 2, 1 i 0**

SELECT nazwa\_wykladu, COUNT(id\_studenta) AS Liczba\_studentow

FROM wyklady w LEFT JOIN studenci\_wyklady sw ON w.id\_wykladu=sw.id\_wykladu

WHERE nazwa\_wykladu LIKE 'Historia%'

GROUP BY w.id\_wykladu, nazwa\_wykladu

ORDER BY Liczba\_studentow DESC

SELECT 3.14

Nazwy wszystkich wykładów z Informatyki oraz średnią ocen ze wszystkich tych wykładów osobno (jeśli jest wiele takich wykładów, to średnia ma być obliczona dla każdego z nich oddzielnie). Jeśli z danego wykładu nie ma żadnej oceny, także powinien on pojawić się na liście. Kolumna ma mieć nazwę Srednia.

**4 rekordy. W trzech ostatnich średnia wynosi NULL**

SELECT nazwa\_wykladu, AVG(ocena) AS Srednia

FROM wyklady w LEFT JOIN oceny\_studentow os ON w.id\_wykladu=os.id\_wykladu

WHERE nazwa\_wykladu='Informatyka'

GROUP BY w.id\_wykladu, nazwa\_wykladu

SELECT 3.15

Nazwy grup, do których jest zapisanych przynajmniej dwóch studentów, liczby studentów zapisanych do tych grup (kolumna ma mieć nazwę LiczbaStudentow) oraz średnie ocen dla każdej grupy (kolumna ma mieć nazwę Srednia). Dane posortowane malejąco według średniej.

**2 rekordy. W pierwszym liczba studentów jest równa 2 w drugim 7**

SELECT nr\_grupy, COUNT(s.id\_studenta) AS LiczbaStudentow, AVG(ocena) AS Srednia

FROM studenci s INNER JOIN oceny\_studentow os ON s.id\_studenta=os.id\_studenta

GROUP BY nr\_grupy

HAVING COUNT(s.id\_studenta) >= 2

ORDER BY Srednia DESC

SELECT 3.16

Nazwiska i imiona wykładowców prowadzących co najmniej jeden wykład wraz z kwotą, jaką uczelnia powinna każdemu z nich wypłacić z tytułu prowadzenia tych wykładów (kwotę za jeden wykład należy obliczyć jako iloczyn stawki godzinowej danego wykładowcy oraz liczby godzin przeznaczonych na wykłady prowadzone przez tego wykładowcę). Pole zawierające kwotę ma mieć nazwę Naleznosci. Dane posortowane rosnąco według nazwiska i imienia.

**4 rekordy. W pierwszym Iwona Halemba, w kolumnie należności kwota 3375**

SELECT nazwisko, imie, SUM(stawka\*liczba\_godzin) AS naleznosci

FROM ((pracownicy INNER JOIN wykladowcy ON id\_pracownika=id\_wykladowcy)

INNER JOIN stopnie\_tytuly st ON st.stopien\_tytul=wykladowcy.stopien\_tytul)

INNER JOIN wyklady w ON w.id\_wykladowcy=wykladowcy.id\_wykladowcy

GROUP BY wykladowcy.id\_wykladowcy, nazwisko, imie

ORDER BY nazwisko, imie

SELECT 3.17

Nazwiska, imiona i nazwy katedr wszystkich wykładowców wraz z liczbą wykładów, które prowadzi każdy z nich. Kolumna zwracająca liczbę wykładów ma mieć nazwę Liczba\_wykladow. Dane posortowane malejąco według liczby wykładów a następnie rosnąco według nazwisk.

**6 rekordów. Najwięcej wykładów prowadzi Jacek Kwiatkowski (6)**

SELECT nazwisko, imie, katedra, COUNT(w.id\_wykladu) AS liczba\_wykladow

FROM (pracownicy INNER JOIN wykladowcy ON id\_pracownika=id\_wykladowcy)

LEFT JOIN wyklady w ON w.id\_wykladowcy=wykladowcy.id\_wykladowcy

GROUP BY wykladowcy.id\_wykladowcy, nazwisko, imie, katedra

ORDER BY liczba\_wykladow DESC, nazwisko

SELECT 3.18

Nazwy wszystkich katedr oraz ile godzin wykładów w sumie mają pracownicy zatrudnieni w tych katedrach. Kolumna zwracająca liczbę godzin ma mieć nazwę Liczba\_godzin. Dane posortowane rosnąco według kolumny Liczba\_godzin.

**4 rekordy. W pierwszym Katedra Ekonomii, 60 godzin**

SELECT katedra, SUM(liczba\_godzin) AS Liczba\_godzin

FROM wykladowcy w LEFT JOIN wyklady wk ON w.id\_wykladowcy=wk.id\_wykladowcy

GROUP BY katedra

ORDER BY Liczba\_godzin

SELECT 3.19

Nazwy wykładów, z których ocena została przyznana co najmniej dwa razy wraz z liczbą przyznanych ocen. Dane posortowane malejąco według nazwy wykładu.

**2 rekordy. Z wykładów ze Statystyki i Historii przyznane zostały po 3 oceny**

SELECT nazwa\_wykladu, count(w.id\_wykladu)

FROM wyklady w INNER JOIN oceny\_studentow os ON w.id\_wykladu=os.id\_wykladu

GROUP BY nazwa\_wykladu, w.id\_wykladu

HAVING COUNT(w.id\_wykladu) >= 2

ORDER BY nazwa\_wykladu DESC