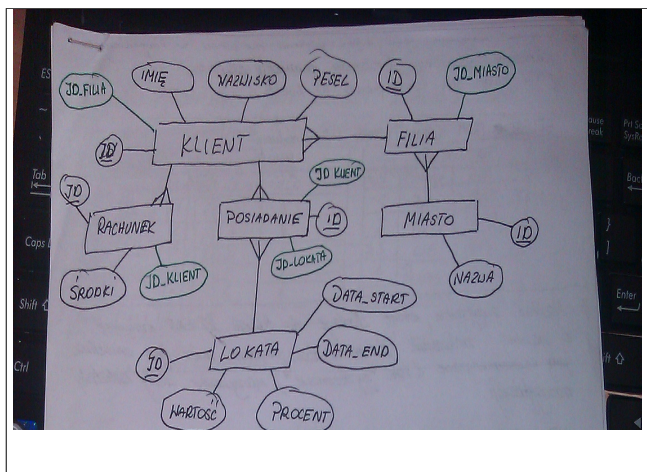


	<pre> CREATE TABLE KLIENT (ID INT PRIMARY KEY, imie varchar(30) not null, nazwisko varchar(30) not null, pesel char(10) not null, JD_FILIA INT REFERENCES FILIA); CREATE TABLE POSIADANIE (JD INT PRIMARY KEY, JD_KLIENT INT, JD_LOKATA INT); CREATE TABLE LOKATA (JD INT PRIMARY KEY, WARTOSC INT, PROCENT INT, DATA_START DATE, DATA_END DATE); CREATE TABLE FILIA (ID INT PRIMARY KEY, ID_MIASTO INT REFERENCES MIASTO); CREATE TABLE KLIENT (ID INTEGER NOT NULL, ID_FILIA INTEGER NOT NULL, IMIE CHARACTER VARYING(30), NAZWISKO CHARACTER VARYING(30), PESEL CHAR(10) NOT NULL, CONSTRAINT KLIENT_PK PRIMARY KEY (ID), CONSTRAINT KLIENT_FILIA_FK FOREIGN KEY (ID_FILIA) REFERENCES FILIA (ID) ON DELETE NO ACTION ON UPDATE NO ACTION); CREATE TABLE POSIADANIE (ID INTEGER NOT NULL, ID_KLIENT INTEGER NOT NULL, ID_LOKATA INTEGER NOT NULL, CONSTRAINT POSIADANIE_KLIENT_FK FOREIGN KEY (ID_KLIENT) REFERENCES KLIENT (ID), CONSTRAINT POSIADANIE_LOKATA_FK FOREIGN KEY (ID_LOKATA) REFERENCES LOKATA (ID)); </pre>
Napisz zapyt SQL które pok 10 lokat o nadl okr realiz (rozn pom datami SELECT * FROM 'lokata' ORDER BY (DATA_END - DATA_START) DESC LIMIT 10;	Nap zap. Które dod do tab. klient kol o nazwie znacznik (typ umozl sys. wpis daty ost mod ALTER TABLE KLIENT ADD ZNACZNIK TIMESTAMP;
Zap, pol liczb lokat (nie wart) wsz lok nal do os nazw S SELECT 'klient' . 'NAZWISKO' , count (lokata.ID) FROM klient, posiadanie, lokata WHERE 'klient' . 'ID' ='posiadanie' . 'ID_klient' AND 'posiadanie' . 'ID_lokata' ='lokata' . 'ID' and 'klient' . 'NAZWISKO' like "s%" group by 'klient' . 'NAZWISKO' ;	Wyj rozn LEFT OUTER JOIN od INNER JOIN SELECT student.nazwisko, przedmiot.nazwa from student inner join stuprz on (stuprz.id_s=student.id) inner join przedmiot on (przedmiot.id=stuprz.id_p); NOWAK - ZARZADZANIE, NOWAK – PROCESY, RYBA - WF, TRELA - WF, MIKS – JAVA SELECT student.nazwisko, przedmiot.nazwa from student left join stuprz on (stuprz.id_s=student.id) join przedmiot on (przedmiot.id=stuprz.id_p); NOWAK-ZARZADZANIE, NOWAK-PROCESY, RYBA-WF, TRELA-WF, SLABY-NULL, KOZA-NULL, MIKS-JAVA
Zapyt które pol srodki na rach (avista) nal do wsz klient (lista klientow i sum) SELECT klient.nazwisko, sum (rachunek.srodki) from klient, rachunek where klient.ID=rachunek.ID_KLIENT group by klient.nazwisko	Zapyt które usuwa filie w miescie SOPOT DELETE FROM filia WHERE ID_MIASTO IN (select ID FROM miasto WHERE nazwa='SOPOT');
Zapyt pod nazwy miast i liczb klient w miastach których liczba przekracza 1000 SELECT miasto.nazwa, count(klient.id) FROM miasto, filia, klient where miato.ID=filia.ID_miasto AND filia.ID=klient.ID_filia group by miasto.nazwa having count (klient.ID)>1000	Param dla funkcji FUNKCION uzytk mogą być typu: IN – IN, OUT i INOUT, INOUT, IN i RETURN, OUT i INOUT
Zapy tw tabl HISTORIA z kolumn data i pesel, trigg kotry po kazdym wst do tabl KLIENT będzie wstawiał do tabl HISTORIA date wpisu i nr pesel wstawianego klienta CREATE TRIGGER dodaj_historie AFTER INSERT ON klient FOR EACH ROW BEGIN INSERT INTO 'historia' VALUES (current_date.data, NEW.pesel); END;;	Ktore z pon zdarzen może uruchomic TRIGGERA wyk zap SELECT, INSERT, UPDATE, zalog się użytkownika
Wskaz poprawna forme instrukcji IF IF imie='Jan' THEN SELECT('Witaj Janku');END IF; IF (imie='Jan') THEN ECHO('Witaj Janku'); END IF; IF imie='Jan' BEGIN SELECT ('Witaj Janku'); END; END IF; IF imie='Jan' BEGIN ECHO ('Witaj Janku'); END; IF (imie IS LIKE „Jan”) THEN SELECT('Witaj Janku');	Dlaczego uzywamy polecenia DELIMITER przy tworzeniu funkcji skladowych: Aby oznaczyc koniec i poczatek funkcji Aby zabezpieczyc się przed blednym wprowadzeniem funkcji w przypadku awarii Aby ustawic znaki konca linii niezaleznie od systemu (linux czy DOS) Bo znak srednika nie może pełnić roli konca instrukcji i znaku terminalnego Żadne z powyższych
Wynikiom działania pol SELECT DATEDIFF(NOW(),2007-12-01) będzie: Liczba TEXT Data NULL Zadne z powyższych	Roznica między HAVING a WHERE polega na tym ze: HAVING sluzy do filtrowania wart zagregowanych a WHERE niezagregowanych WHERE wymaga zastosowania GROUP BY WHERE sluzy do filtrowania wart zagregowanych a HAVING niezagregowanych HAVING wymaga zastosowania COUNT lub SUM zadne z powyższych
Aby wyswietlic pierwsze 10 rekordow zapytania należy wykonac: SELECT * FROM student FIRST 10 SELECT 10, * FROM student SELECT * FROM student LIMIT FIRST 10 SELECT 10 FROM student SELECT * FROM student LIMIT 10	Unia UNION sluzy w SQL do: Tworzenia zlaczon tabel (tzw UNION JOIN) Łączenia wyników różnych zapytan w jedną tabelę Konwersji różnych zmiennych do jednego typu tworzenia plikow unii na dysku zadne z powyższych
Na pewnej tablicy zalożono trigger ON INSERT zadziałal on w nast. Sytuacjach W przypadku wykonania LOAD DATA INFILE W przypadku wykonania UPDATE na kolumnie typu TIMESTAMP W przypadku wykonania DROP TABLE Wszystkie odpowiedzi (A-C) są poprawne Zadne z powyższych	Które z poniższych typow zlaczenia JOIN nie istnieje w SQL (niekoniecznie w MySQL) NATURAL JOIN FULL OUTER JOIN LEFT OUTER JOIN FULL INNER JOIN wszystkie istnieja
Typ danych BLOB uzywany jest do: przechowywania logow baz danych przechowywania obiektow tekstowych większych niż 255 znakow przechowywania danych zaszyfrowanych fukcja BLOWFISH przechowywania danych binarnych zadne z powyższych	Transakcje to mechanizm zabezpieczajacy przed: Niewykonaniem się kilku polecen Powstaniem niespojnosci referencyjnej Blednym naliczeniem funkcji matematycznych Bledami indeksowania Wykonaniem się czesci polecen i niewykonaniem się pozostalych
Zastosowanie indeksow: Przyspiesza działanie SELECT, ale zwalnia UPDATE, DELETE i INSERT Przyspiesza działanie SELECT i UPDATE, ale zwalnia DELETE i INSERT Przyspiesza działanie SELECT UPDATE DELETE i INSERT Przyspiesza działanie UPDATE DELETE i INSERT, ale zwalnia SELECT Zadne z powyższych	