Szymon Adamczyk 264456

Dawid Makielak 264003

Daniel Ryszkowski 266453

Sprawozdanie Bazy danych Projekt

Etap 2

**1. Identyfikacja Encji:**

1. Dostawcy

2. Kategorie

3. Klienci

4. ListaAdministratorow

5. ListaDostawcow

6. ListaDostawcow\_Dostawcy

7. ListaMagazynow

8. Magazyn

9. OpinieKlientow

10. Pracownicy

11. Produkty

12. Produkty\_Kategorie

13. Transakcje

14. Zamówienia

15. Zamówienia\_Transakcje

**2. Sformułowanie Wymagań:**

1. Sprzedawca:

• Zamówienia: Wyszukiwanie

• Magazyn: Wyszukiwanie, modyfikacja

• Produkty: Wyszukiwanie

• Produkty\_Kategorie: Wyszukiwanie

• Kategorie: Wyszukiwanie

• OpinieKlientow: Wyszukiwanie

• Klienci: Wyszukiwanie

2. Handlowiec:

• Zamówienia: Wyszukiwanie, wstawianie, modyfikacja, usuwanie

• Magazyn: Wyszukiwanie, wstawianie, modyfikacja, usuwanie

• Dostawcy: Wyszukiwanie

• Produkty: Wyszukiwanie, wstawianie, modyfikacja, usuwanie

• Transakcje: Wyszukiwanie, wstawianie, modyfikacja, usuwanie

• Klienci: Wyszukiwanie

• Kategorie: Wyszukiwanie

• OpinieKlientow: Wyszukiwanie

• Klienci: Wyszukiwanie

3. Administrator:

• Wszystko: Wstawianie, modyfikacja, usuwanie, wyszukiwanie

**3. Diagram konceptualny:**

**Obraz zawierający diagram, tekst, Plan, Karteczka samoprzylepna

Opis wygenerowany automatycznie**

**4. Diagram Fizyczny**

Obraz zawierający tekst, zrzut ekranu, list, Karteczka samoprzylepna

Opis wygenerowany automatycznie

**5. Fizyczna baza danych:**

**a) Skrypt do generowania bazy:**

|  |
| --- |
| CREATE TABLE Dostawcy (IdDostawcy int(10) NOT NULL AUTO\_INCREMENT, NazwaDostawcy varchar(255) NOT NULL, NumerTelefonu varchar(10) NOT NULL, PRIMARY KEY (IdDostawcy));  CREATE TABLE Kategorie (NazwaKategorii varchar(255) NOT NULL, OpisKategorii varchar(65000) NOT NULL, PRIMARY KEY (NazwaKategorii));  CREATE TABLE Klienci (nick varchar(20) NOT NULL, Imie varchar(50) NOT NULL, Nazwisko varchar(50) NOT NULL, Adres varchar(50) NOT NULL, Email varchar(60) NOT NULL UNIQUE, NumerTelefonu varchar(10), NumerTransakcji varchar(10) NOT NULL, PRIMARY KEY (nick));  CREATE TABLE ListaAdministratorow (AdministratorId int(10) NOT NULL AUTO\_INCREMENT, Imie varchar(255) NOT NULL, Nazwisko varchar(255) NOT NULL, DataZatrudnienia date NOT NULL, PRIMARY KEY (AdministratorId));  CREATE TABLE ListaDostawcow (IdListy int(10) NOT NULL AUTO\_INCREMENT, IdDostawcy int(10) NOT NULL, IdMagazynu int(10) NOT NULL, PRIMARY KEY (IdListy));  CREATE TABLE ListaMagazynow (IdMagazynu int(10) NOT NULL AUTO\_INCREMENT, Adres varchar(50) NOT NULL, AdministratorId int(10) NOT NULL, PRIMARY KEY (IdMagazynu));  CREATE TABLE Magazyn (IdMagazynu int(10) NOT NULL, IdProduktu int(10) NOT NULL, Ilosc int(10) NOT NULL, PRIMARY KEY (IdMagazynu));  CREATE TABLE OpinieKlientow (Ocena int(1), Komentarz varchar(650000), nick varchar(20) NOT NULL, IdProduktu int(10) NOT NULL);  CREATE TABLE Pracownicy (Id int(10) NOT NULL AUTO\_INCREMENT, Imie varchar(50) NOT NULL, Nazwisko varchar(50) NOT NULL, Stanowisko varchar(20) NOT NULL, DataZatrudnienia date NOT NULL, AdministratorId int(10) NOT NULL, PRIMARY KEY (Id));  CREATE TABLE Produkty (IdProduktu int(10) NOT NULL AUTO\_INCREMENT, NazwaProduktu varchar(255) NOT NULL, Cena float NOT NULL, Opis varchar(65000) NOT NULL, Kategoria varchar(255) NOT NULL, PRIMARY KEY (IdProduktu));  CREATE TABLE Transakcje (NumerTransakcji varchar(20) NOT NULL, DataTransakcji date NOT NULL, Kwota int(10) NOT NULL, NumerZamowienia varchar(10), PRIMARY KEY (NumerTransakcji));  CREATE TABLE Zamówienia (NumerZamowienia varchar(10) NOT NULL, DataTransakcji date NOT NULL, StanZamowienia varchar(255) NOT NULL, PRIMARY KEY (NumerZamowienia));  CREATE TABLE Zamówienia\_Transakcje (ZamówieniaNumerZamowienia varchar(10) NOT NULL, TransakcjeNumerTransakcji varchar(20) NOT NULL, DataTransakcji date NOT NULL, PRIMARY KEY (ZamówieniaNumerZamowienia, TransakcjeNumerTransakcji));  ALTER TABLE ListaDostawcow ADD CONSTRAINT FKListaDosta431852 FOREIGN KEY (IdMagazynu) REFERENCES ListaMagazynow (IdMagazynu);  ALTER TABLE ListaDostawcow ADD CONSTRAINT FKListaDosta626501 FOREIGN KEY (IdDostawcy) REFERENCES Dostawcy (IdDostawcy);  ALTER TABLE Produkty ADD CONSTRAINT FKProdukty596731 FOREIGN KEY (Kategoria) REFERENCES Kategorie (NazwaKategorii);  ALTER TABLE Transakcje ADD CONSTRAINT FKTransakcje585185 FOREIGN KEY (NumerTransakcji) REFERENCES Klienci (nick);  ALTER TABLE OpinieKlientow ADD CONSTRAINT FKOpinieKlie787508 FOREIGN KEY (nick) REFERENCES Klienci (nick);  ALTER TABLE OpinieKlientow ADD CONSTRAINT FKOpinieKlie85286 FOREIGN KEY (IdProduktu) REFERENCES Produkty (IdProduktu);  ALTER TABLE Magazyn ADD CONSTRAINT FKMagazyn299317 FOREIGN KEY (IdProduktu) REFERENCES Produkty (IdProduktu);  ALTER TABLE Magazyn ADD CONSTRAINT FKMagazyn106861 FOREIGN KEY (IdMagazynu) REFERENCES ListaMagazynow (IdMagazynu);  ALTER TABLE Pracownicy ADD CONSTRAINT FKPracownicy675238 FOREIGN KEY (AdministratorId) REFERENCES ListaAdministratorow (AdministratorId);  ALTER TABLE ListaMagazynow ADD CONSTRAINT FKListaMagaz601202 FOREIGN KEY (AdministratorId) REFERENCES ListaAdministratorow (AdministratorId);  ALTER TABLE Zamówienia\_Transakcje ADD CONSTRAINT FKZamówienia498578 FOREIGN KEY (ZamówieniaNumerZamowienia) REFERENCES Zamówienia (NumerZamowienia);  ALTER TABLE Zamówienia\_Transakcje ADD CONSTRAINT FKZamówienia973464 FOREIGN KEY (TransakcjeNumerTransakcji) REFERENCES Transakcje (NumerTransakcji); |

**b) Skrypt do generowania użytkowników:**

|  |
| --- |
| CREATE USER 'Sprzedawca'@'localhost' IDENTIFIED BY 'haslo\_sprzedawcy';  GRANT SELECT ON baza\_danych.Zamówienia TO 'Sprzedawca'@'localhost';  GRANT SELECT, UPDATE ON baza\_danych.Magazyn TO 'Sprzedawca'@'localhost';  GRANT SELECT ON baza\_danych.Produkty TO 'Sprzedawca'@'localhost';  GRANT SELECT ON baza\_danych.Produkty\_Kategorie TO 'Sprzedawca'@'localhost';  GRANT SELECT ON baza\_danych.Kategorie TO 'Sprzedawca'@'localhost';  GRANT SELECT ON baza\_danych.OpinieKlientow TO 'Sprzedawca'@'localhost';  GRANT SELECT ON baza\_danych.Klienci TO 'Sprzedawca'@'localhost';  CREATE USER 'Handlowiec'@'localhost' IDENTIFIED BY 'haslo\_handlowca';  GRANT SELECT, INSERT, UPDATE, DELETE ON baza\_danych.Zamówienia TO 'Handlowiec'@'localhost';  GRANT SELECT, INSERT, UPDATE, DELETE ON baza\_danych.Magazyn TO 'Handlowiec'@'localhost';  GRANT SELECT ON baza\_danych.Dostawcy TO 'Handlowiec'@'localhost';  GRANT SELECT, INSERT, UPDATE, DELETE ON baza\_danych.Produkty TO 'Handlowiec'@'localhost';  GRANT SELECT, INSERT, UPDATE, DELETE ON baza\_danych.Transakcje TO 'Handlowiec'@'localhost';  GRANT SELECT ON baza\_danych.Klienci TO 'Handlowiec'@'localhost';  GRANT SELECT ON baza\_danych.Kategorie TO 'Handlowiec'@'localhost';  GRANT SELECT ON baza\_danych.OpinieKlientow TO 'Handlowiec'@'localhost';  CREATE USER 'Administrator'@'localhost' IDENTIFIED BY 'haslo\_administratora';  GRANT ALL PRIVILEGES ON baza\_danych.\* TO 'Administrator'@'localhost'; |

**c) Funkcje:**

|  |
| --- |
| DELIMITER //  CREATE PROCEDURE DodajDostawce (  IN p\_NazwaDostawcy VARCHAR(255),  IN p\_NumerTelefonu VARCHAR(10),  IN p\_IdMagazynu INT  )  BEGIN  INSERT INTO Dostawcy (NazwaDostawcy, NumerTelefonu, IdMagazynu)  VALUES (p\_NazwaDostawcy, p\_NumerTelefonu, p\_IdMagazynu);  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE UsunDostawce (  IN p\_IdDostawcy INT  )  BEGIN  DELETE FROM Dostawcy WHERE IdDostawcy = p\_IdDostawcy;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE AktualizujDostawce (  IN p\_IdDostawcy INT,  IN p\_NazwaDostawcy VARCHAR(255),  IN p\_NumerTelefonu VARCHAR(10),  IN p\_IdMagazynu INT  )  BEGIN  UPDATE Dostawcy  SET NazwaDostawcy = p\_NazwaDostawcy,  NumerTelefonu = p\_NumerTelefonu,  IdMagazynu = p\_IdMagazynu  WHERE IdDostawcy = p\_IdDostawcy;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE DodajKategorieProduktu(  IN in\_ProduktyId int(10),  IN in\_KategorieNazwa varchar(255)  )  BEGIN  INSERT INTO Produkty\_Kategorie (ProduktyId, KategorieNazwaKategorii)  VALUES (in\_ProduktyId, in\_KategorieNazwa);  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE UsunKategorieProduktu(  IN in\_ProduktyId int(10),  IN in\_KategorieNazwa varchar(255)  )  BEGIN  DELETE FROM Produkty\_Kategorie  WHERE ProduktyId = in\_ProduktyId AND KategorieNazwaKategorii = in\_KategorieNazwa;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE PobierzKategorieProduktu(  IN in\_ProduktyId int(10)  )  BEGIN  SELECT \* FROM Produkty\_Kategorie WHERE ProduktyId = in\_ProduktyId;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE EdytujKategorie (  IN p\_IdKategorii INT,  IN p\_NazwaKategorii VARCHAR(255)  )  BEGIN  UPDATE Kategorie  SET NazwaKategorii = p\_NazwaKategorii  WHERE IdKategorii = p\_IdKategorii;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE DodajKlienta(  IN in\_nick varchar(20),  IN in\_Imie varchar(50),  IN in\_Nazwisko varchar(50),  IN in\_Adres varchar(50),  IN in\_Email varchar(60),  IN in\_NumerTelefonu varchar(10),  IN in\_NumerTransakcji varchar(10)  )  BEGIN  INSERT INTO Klienci (nick, Imie, Nazwisko, Adres, Email, NumerTelefonu, NumerTransakcji)  VALUES (in\_nick, in\_Imie, in\_Nazwisko, in\_Adres, in\_Email, in\_NumerTelefonu, in\_NumerTransakcji);  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE AktualizujKlienta(  IN in\_nick varchar(20),  IN in\_Imie varchar(50),  IN in\_Nazwisko varchar(50),  IN in\_Adres varchar(50),  IN in\_Email varchar(60),  IN in\_NumerTelefonu varchar(10),  IN in\_NumerTransakcji varchar(10)  )  BEGIN  UPDATE Klienci  SET  Imie = in\_Imie,  Nazwisko = in\_Nazwisko,  Adres = in\_Adres,  Email = in\_Email,  NumerTelefonu = in\_NumerTelefonu,  NumerTransakcji = in\_NumerTransakcji  WHERE nick = in\_nick;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE UsunKlienta(IN in\_nick varchar(20))  BEGIN  DELETE FROM Klienci WHERE nick = in\_nick;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE PobierzDaneKlienta(IN in\_nick varchar(20))  BEGIN  SELECT \* FROM Klienci WHERE nick = in\_nick;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE DodajAdministratora(  IN in\_Imie varchar(255),  IN in\_Nazwisko varchar(255),  IN in\_DataZatrudnienia date  )  BEGIN  INSERT INTO ListaAdministratorow (Imie, Nazwisko, DataZatrudnienia)  VALUES (in\_Imie, in\_Nazwisko, in\_DataZatrudnienia);  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE AktualizujAdministratora(  IN in\_AdministratorId int(10),  IN in\_Imie varchar(255),  IN in\_Nazwisko varchar(255),  IN in\_DataZatrudnienia date  )  BEGIN  UPDATE ListaAdministratorow  SET  Imie = in\_Imie,  Nazwisko = in\_Nazwisko,  DataZatrudnienia = in\_DataZatrudnienia  WHERE AdministratorId = in\_AdministratorId;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE UsunAdministratora(IN in\_AdministratorId int(10))  BEGIN  DELETE FROM ListaAdministratorow WHERE AdministratorId = in\_AdministratorId;  END //  DELIMITER ;  CREATE PROCEDURE PobierzIloscProduktuWMagazynie(  IN in\_IdMagazynu int(10),  IN in\_IdProduktu int(10)  )  BEGIN  SELECT Ilosc FROM Magazyn  WHERE IdMagazynu = in\_IdMagazynu AND IdProduktu = in\_IdProduktu;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE UsunProduktZMagazynu(  IN in\_IdMagazynu int(10),  IN in\_IdProduktu int(10)  )  BEGIN  DELETE FROM Magazyn  WHERE IdMagazynu = in\_IdMagazynu AND IdProduktu = in\_IdProduktu;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE AktualizujIloscProduktuWMagazynie(  IN in\_IdMagazynu int(10),  IN in\_IdProduktu int(10),  IN in\_NowaIlosc int(10)  )  BEGIN  UPDATE Magazyn  SET Ilosc = in\_NowaIlosc  WHERE IdMagazynu = in\_IdMagazynu AND IdProduktu = in\_IdProduktu;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE DodajProduktDoMagazynu(  IN in\_IdMagazynu int(10),  IN in\_IdProduktu int(10),  IN in\_Ilosc int(10)  )  BEGIN  INSERT INTO Magazyn (IdMagazynu, IdProduktu, Ilosc)  VALUES (in\_IdMagazynu, in\_IdProduktu, in\_Ilosc);  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE DodajOpinie(  IN in\_Ocena int(1),  IN in\_Komentarz varchar(10000),  IN in\_nick varchar(20),  IN in\_IdProduktu int(10)  )  BEGIN  INSERT INTO OpinieKlientow (Ocena, Komentarz, nick, IdProduktu)  VALUES (in\_Ocena, in\_Komentarz, in\_nick, in\_IdProduktu);  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE AktualizujOpinie(  IN in\_Ocena int(1),  IN in\_Komentarz varchar(10000),  IN in\_nick varchar(20),  IN in\_IdProduktu int(10)  )  BEGIN  UPDATE OpinieKlientow  SET Ocena = in\_Ocena,  Komentarz = in\_Komentarz  WHERE nick = in\_nick AND IdProduktu = in\_IdProduktu;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE UsunOpinie(  IN in\_nick varchar(20),  IN in\_IdProduktu int(10)  )  BEGIN  DELETE FROM OpinieKlientow  WHERE nick = in\_nick AND IdProduktu = in\_IdProduktu;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE PobierzOpinieProduktu(  IN in\_IdProduktu int(10)  )  BEGIN  SELECT \* FROM OpinieKlientow WHERE IdProduktu = in\_IdProduktu;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE DodajPracownika(  IN in\_Imie varchar(50),  IN in\_Nazwisko varchar(50),  IN in\_Stanowisko varchar(20),  IN in\_DataZatrudnienia date,  IN in\_AdministratorId int(10)  )  BEGIN  INSERT INTO Pracownicy (Imie, Nazwisko, Stanowisko, DataZatrudnienia, AdministratorId)  VALUES (in\_Imie, in\_Nazwisko, in\_Stanowisko, in\_DataZatrudnienia, in\_AdministratorId);  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE AktualizujPracownika(  IN in\_Id int(10),  IN in\_Imie varchar(50),  IN in\_Nazwisko varchar(50),  IN in\_Stanowisko varchar(20),  IN in\_DataZatrudnienia date,  IN in\_AdministratorId int(10)  )  BEGIN  UPDATE Pracownicy  SET  Imie = in\_Imie,  Nazwisko = in\_Nazwisko,  Stanowisko = in\_Stanowisko,  DataZatrudnienia = in\_DataZatrudnienia,  AdministratorId = in\_AdministratorId  WHERE Id = in\_Id;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE UsunPracownika(  IN in\_Id int(10)  )  BEGIN  DELETE FROM Pracownicy WHERE Id = in\_Id;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE PobierzDanePracownika(  IN in\_Id int(10)  )  BEGIN  SELECT \* FROM Pracownicy WHERE Id = in\_Id;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE AktualizujProdukt(  IN in\_IdProduktu int(10),  IN in\_NazwaProduktu varchar(255),  IN in\_Cena float,  IN in\_Opis varchar(10000),  IN in\_Kategoria varchar(255)  )  BEGIN  UPDATE Produkty  SET  NazwaProduktu = in\_NazwaProduktu,  Cena = in\_Cena,  Opis = in\_Opis,  Kategoria = in\_Kategoria  WHERE IdProduktu = in\_IdProduktu;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE UsunProdukt(  IN in\_IdProduktu int(10)  )  BEGIN  DELETE FROM Produkty WHERE IdProduktu = in\_IdProduktu;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE PobierzDaneProduktu(  IN in\_IdProduktu int(10)  )  BEGIN  SELECT \* FROM Produkty WHERE IdProduktu = in\_IdProduktu;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE DodajTransakcje(  IN in\_NumerTransakcji varchar(10),  IN in\_DataTransakcji date,  IN in\_Kwota int(10),  IN in\_NumerZamowienia varchar(10)  )  BEGIN  INSERT INTO Transakcje (NumerTransakcji, DataTransakcji, Kwota, NumerZamowienia)  VALUES (in\_NumerTransakcji, in\_DataTransakcji, in\_Kwota, in\_NumerZamowienia);  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE AktualizujTransakcje(  IN in\_NumerTransakcji varchar(10),  IN in\_NowyNumerZamowienia varchar(10)  )  BEGIN  UPDATE Transakcje  SET NumerZamowienia = in\_NowyNumerZamowienia  WHERE NumerTransakcji = in\_NumerTransakcji;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE UsunTransakcje(  IN in\_NumerTransakcji varchar(10)  )  BEGIN  DELETE FROM Transakcje WHERE NumerTransakcji = in\_NumerTransakcji;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE PobierzDaneTransakcji(  IN in\_NumerTransakcji varchar(10)  )  BEGIN  SELECT \* FROM Transakcje WHERE NumerTransakcji = in\_NumerTransakcji;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE PobierzDaneZamowienia(  IN in\_NumerZamowienia varchar(10)  )  BEGIN  SELECT \* FROM Zamówienia WHERE NumerZamowienia = in\_NumerZamowienia;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE AktualizujStanZamowienia(  IN in\_NumerZamowienia varchar(10),  IN in\_NowyStanZamowienia varchar(255)  )  BEGIN  UPDATE Zamówienia  SET StanZamowienia = in\_NowyStanZamowienia  WHERE NumerZamowienia = in\_NumerZamowienia;  END //  DELIMITER ;  DELIMITER //  CREATE PROCEDURE DodajZamowienie(  IN in\_NumerZamowienia varchar(10),  IN in\_DataTransakcji date,  IN in\_StanZamowienia varchar(255)  )  BEGIN  INSERT INTO Zamówienia (NumerZamowienia, DataTransakcji, StanZamowienia)  VALUES (in\_NumerZamowienia, in\_DataTransakcji, in\_StanZamowienia);  END //  DELIMITER ; |

**d) Trigery:**

|  |  |
| --- | --- |
| DELIMITER //  CREATE TRIGGER before\_delete\_Magazyn  BEFORE DELETE ON Magazyn  FOR EACH ROW  BEGIN  DELETE FROM ListaMagazynow  WHERE IdMagazynu = OLD.IdMagazynu;  END;  //  DELIMITER ;  DELIMITER //  CREATE TRIGGER before\_delete\_Dostawcy  BEFORE DELETE ON Dostawcy  FOR EACH ROW  BEGIN  DELETE FROM ListaDostawcow  WHERE IdDostawcy = OLD.IdDostawcy;  END;  //  DELIMITER ; |  |

sprawdzanie czy wszystkie pola sa wypelnione

|  |
| --- |
| CREATE TRIGGER check\_not\_null  BEFORE INSERT OR UPDATE ON Klienci  FOR EACH ROW  BEGIN  IF NEW.nick IS NULL OR NEW.Imie IS NULL OR NEW.Nazwisko IS NULL OR NEW.Email IS NULL THEN  SIGNAL SQLSTATE '45000'  SET MESSAGE\_TEXT = 'Wszystkie pola klienta muszą być wypełnione.';  END IF;  END; |

Sprawdzanie czy nazwa produktu nie jest już w bazie danych

|  |
| --- |
| CREATE TRIGGER check\_product  BEFORE INSERT ON Produkty  FOR EACH ROW  BEGIN  IF EXISTS (SELECT 1 FROM Produkty WHERE NazwaProduktu = NEW.NazwaProduktu) THEN  SIGNAL SQLSTATE '45000'  SET MESSAGE\_TEXT = 'Produkt o tej nazwie już istnieje.';  END IF;  END; |

Sprawdzanie poprawności danych

|  |
| --- |
| CREATE TRIGGER check\_date  BEFORE INSERT ON Transakcje  FOR EACH ROW  BEGIN  IF NEW.DataTransakcji > CURRENT\_DATE() THEN  SIGNAL SQLSTATE '45000'  SET MESSAGE\_TEXT = 'Data transakcji nie może być w przyszłości.';  END IF;  END; |

**6. Testy**

**Implementacja Bazy danych**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 62 | 17:18:49 | CREATE TABLE Dostawcy (IdDostawcy int(10) NOT NULL AUTO\_INCREMENT, NazwaDostawcy varchar(255) NOT NULL, NumerTelefonu varchar(10) NOT NULL, PRIMARY KEY (IdDostawcy)) | 0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. | 0.016 sec |
| 63 | 17:18:49 | CREATE TABLE Kategorie (NazwaKategorii varchar(255) NOT NULL, OpisKategorii varchar(15000) NOT NULL, PRIMARY KEY (NazwaKategorii)) | 0 row(s) affected | 0.016 sec |
| 64 | 17:18:49 | CREATE TABLE Klienci (nick varchar(20) NOT NULL, Imie varchar(50) NOT NULL, Nazwisko varchar(50) NOT NULL, Adres varchar(50) NOT NULL, Email varchar(60) NOT NULL UNIQUE, NumerTelefonu varchar(10), NumerTransakcji varchar(10) NOT NULL, PRIMARY KEY (nick)) | 0 row(s) affected | 0.015 sec |
| 65 | 17:18:49 | CREATE TABLE ListaAdministratorow (AdministratorId int(10) NOT NULL AUTO\_INCREMENT, Imie varchar(255) NOT NULL, Nazwisko varchar(255) NOT NULL, DataZatrudnienia date NOT NULL, PRIMARY KEY (AdministratorId)) | 0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. | 0.016 sec |
| 66 | 17:18:49 | CREATE TABLE ListaDostawcow (IdListy int(10) NOT NULL AUTO\_INCREMENT, IdDostawcy int(10) NOT NULL, IdMagazynu int(10) NOT NULL, PRIMARY KEY (IdListy)) | 0 row(s) affected, 3 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. 1681 Integer display width is deprecated and will be removed in a future release. 1681 Integer display width is deprecated and will be removed in a future release. | 0.015 sec |
| 67 | 17:18:49 | CREATE TABLE ListaMagazynow (IdMagazynu int(10) NOT NULL AUTO\_INCREMENT, Adres varchar(50) NOT NULL, AdministratorId int(10) NOT NULL, PRIMARY KEY (IdMagazynu)) | 0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. 1681 Integer display width is deprecated and will be removed in a future release. | 0.016 sec |
| 68 | 17:18:49 | CREATE TABLE Magazyn (IdMagazynu int(10) NOT NULL, IdProduktu int(10) NOT NULL, Ilosc int(10) NOT NULL, PRIMARY KEY (IdMagazynu)) | 0 row(s) affected, 3 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. 1681 Integer display width is deprecated and will be removed in a future release. 1681 Integer display width is deprecated and will be removed in a future release. | 0.000 sec |
| 69 | 17:18:49 | CREATE TABLE OpinieKlientow (Ocena int(1), Komentarz varchar(15000), nick varchar(20) NOT NULL, IdProduktu int(10) NOT NULL) | 0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. 1681 Integer display width is deprecated and will be removed in a future release. | 0.016 sec |
| 70 | 17:18:49 | CREATE TABLE Pracownicy (Id int(10) NOT NULL AUTO\_INCREMENT, Imie varchar(50) NOT NULL, Nazwisko varchar(50) NOT NULL, Stanowisko varchar(20) NOT NULL, DataZatrudnienia date NOT NULL, AdministratorId int(10) NOT NULL, PRIMARY KEY (Id)) | 0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. 1681 Integer display width is deprecated and will be removed in a future release. | 0.015 sec |
| 71 | 17:18:49 | CREATE TABLE Produkty (IdProduktu int(10) NOT NULL AUTO\_INCREMENT, NazwaProduktu varchar(255) NOT NULL, Cena float NOT NULL, Opis varchar(15000) NOT NULL, Kategoria varchar(255) NOT NULL, PRIMARY KEY (IdProduktu)) | 0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. | 0.016 sec |
| 72 | 17:18:49 | CREATE TABLE Transakcje (NumerTransakcji varchar(20) NOT NULL, DataTransakcji date NOT NULL, Kwota int(10) NOT NULL, NumerZamowienia varchar(10), PRIMARY KEY (NumerTransakcji)) | 0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. | 0.016 sec |
| 73 | 17:18:49 | CREATE TABLE Zamówienia (NumerZamowienia varchar(10) NOT NULL, DataTransakcji date NOT NULL, StanZamowienia varchar(255) NOT NULL, PRIMARY KEY (NumerZamowienia)) | 0 row(s) affected | 0.015 sec |
| 74 | 17:18:49 | CREATE TABLE Zamówienia\_Transakcje (ZamówieniaNumerZamowienia varchar(10) NOT NULL, TransakcjeNumerTransakcji varchar(20) NOT NULL, DataTransakcji date NOT NULL, PRIMARY KEY (ZamówieniaNumerZamowienia, TransakcjeNumerTransakcji)) | 0 row(s) affected | 0.000 sec |
| 75 | 17:18:49 | ALTER TABLE ListaDostawcow ADD CONSTRAINT FKListaDosta431852 FOREIGN KEY (IdMagazynu) REFERENCES ListaMagazynow (IdMagazynu) | 0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0 | 0.031 sec |
| 76 | 17:18:49 | ALTER TABLE ListaDostawcow ADD CONSTRAINT FKListaDosta626501 FOREIGN KEY (IdDostawcy) REFERENCES Dostawcy (IdDostawcy) | 0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0 | 0.032 sec |
| 77 | 17:18:49 | ALTER TABLE Produkty ADD CONSTRAINT FKProdukty596731 FOREIGN KEY (Kategoria) REFERENCES Kategorie (NazwaKategorii) | 0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0 | 0.047 sec |
| 78 | 17:18:50 | ALTER TABLE Transakcje ADD CONSTRAINT FKTransakcje585185 FOREIGN KEY (NumerTransakcji) REFERENCES Klienci (nick) | 0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0 | 0.015 sec |
| 79 | 17:18:50 | ALTER TABLE OpinieKlientow ADD CONSTRAINT FKOpinieKlie787508 FOREIGN KEY (nick) REFERENCES Klienci (nick) | 0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0 | 0.031 sec |
| 80 | 17:18:50 | ALTER TABLE OpinieKlientow ADD CONSTRAINT FKOpinieKlie85286 FOREIGN KEY (IdProduktu) REFERENCES Produkty (IdProduktu) | 0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0 | 0.032 sec |
| 81 | 17:18:50 | ALTER TABLE Magazyn ADD CONSTRAINT FKMagazyn299317 FOREIGN KEY (IdProduktu) REFERENCES Produkty (IdProduktu) | 0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0 | 0.032 sec |
| 82 | 17:18:50 | ALTER TABLE Magazyn ADD CONSTRAINT FKMagazyn106861 FOREIGN KEY (IdMagazynu) REFERENCES ListaMagazynow (IdMagazynu) | 0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0 | 0.031 sec |
| 83 | 17:18:50 | ALTER TABLE Pracownicy ADD CONSTRAINT FKPracownicy675238 FOREIGN KEY (AdministratorId) REFERENCES ListaAdministratorow (AdministratorId) | 0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0 | 0.015 sec |
| 84 | 17:18:50 | ALTER TABLE ListaMagazynow ADD CONSTRAINT FKListaMagaz601202 FOREIGN KEY (AdministratorId) REFERENCES ListaAdministratorow (AdministratorId) | 0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0 | 0.031 sec |
| 85 | 17:18:50 | ALTER TABLE Zamówienia\_Transakcje ADD CONSTRAINT FKZamówienia498578 FOREIGN KEY (ZamówieniaNumerZamowienia) REFERENCES Zamówienia (NumerZamowienia) | 0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0 | 0.016 sec |
| 86 | 17:18:50 | ALTER TABLE Zamówienia\_Transakcje ADD CONSTRAINT FKZamówienia973464 FOREIGN KEY (TransakcjeNumerTransakcji) REFERENCES Transakcje (NumerTransakcji) | 0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0 | 0.031 sec |

**Testy integracji semantycznej na przykładach:**

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, numer

Opis wygenerowany automatycznie

**Sprawdzanie Infernalności encji:**

-Sprawdzenie czy klucze głowne sa unikalne i niepuste

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Opis wygenerowany automatycznie

-Sprawdzenie czy nie występują duplikaty kluczy głównych dla ListaMagazynow

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

**Sprawdzenie integralności referencji**

Sprawdź, czy wszystkie klucze obce są poprawne:

|  |
| --- |
| SELECT \* FROM ListaDostawcow  WHERE IdMagazynu NOT IN (SELECT IdMagazynu FROM ListaMagazynow);  SELECT \* FROM ListaDostawcow  WHERE IdDostawcy NOT IN (SELECT IdDostawcy FROM Dostawcy);  SELECT \* FROM Produkty  WHERE Kategoria NOT IN (SELECT NazwaKategorii FROM Kategorie);  SELECT \* FROM Transakcje  WHERE NumerTransakcji NOT IN (SELECT nick FROM Klienci);  SELECT \* FROM OpinieKlientow  WHERE nick NOT IN (SELECT nick FROM Klienci) OR  IdProduktu NOT IN (SELECT IdProduktu FROM Produkty);  SELECT \* FROM Magazyn  WHERE IdProduktu NOT IN (SELECT IdProduktu FROM Produkty) OR  IdMagazynu NOT IN (SELECT IdMagazynu FROM ListaMagazynow);  SELECT \* FROM Pracownicy  WHERE AdministratorId NOT IN (SELECT AdministratorId FROM ListaAdministratorow);  SELECT \* FROM ListaMagazynow  WHERE AdministratorId NOT IN (SELECT AdministratorId FROM ListaAdministratorow);  SELECT \* FROM Zamówienia\_Transakcje  WHERE ZamówieniaNumerZamowienia NOT IN (SELECT NumerZamowienia FROM Zamówienia) OR  TransakcjeNumerTransakcji NOT IN (SELECT NumerTransakcji FROM Transakcje); |

Powyższe zapytanie powinno nie zwrócić nam żadnych wyników co oznacza ze klucze obce są poprawne:  
Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie