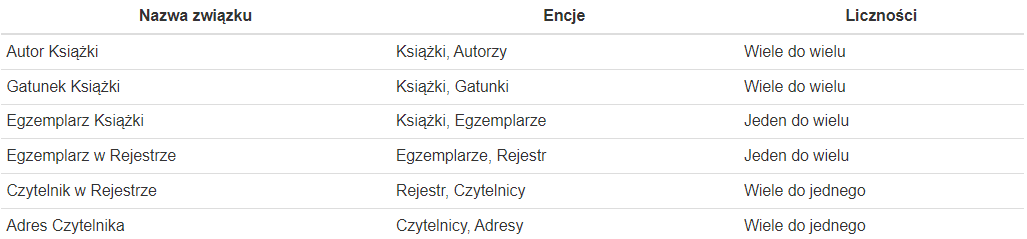
Raport z bazy danych - Biblioteka

Jan Moskal i Szymon Makulec

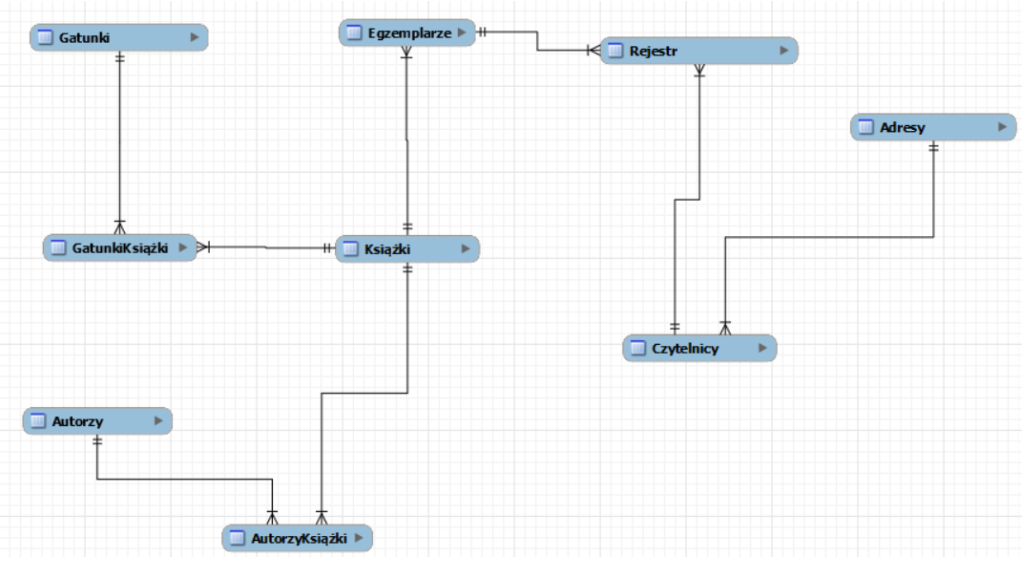
Tematem niniejszego projektu jest

Wyniki etapu 1

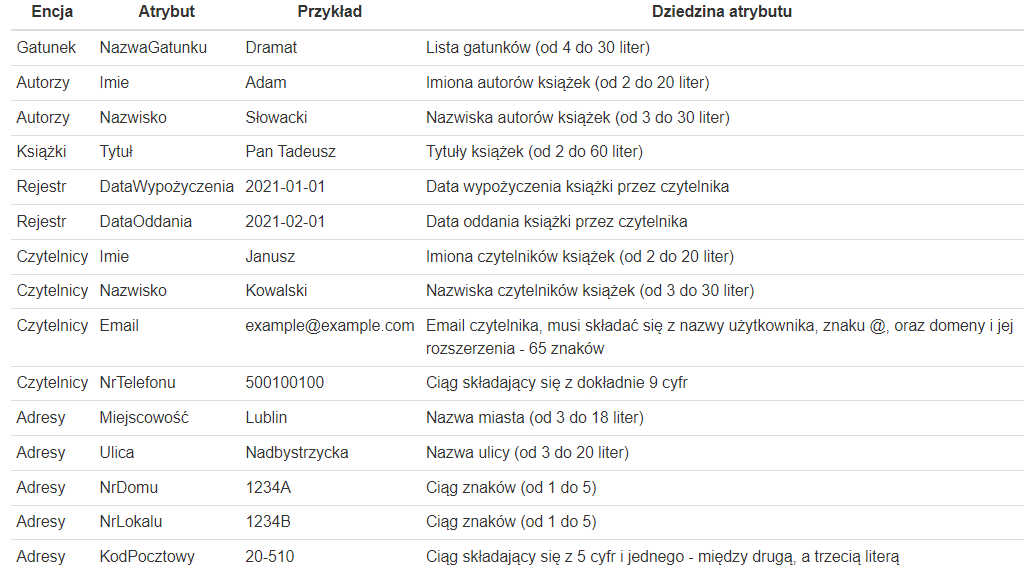
## Wyniki etapu 2



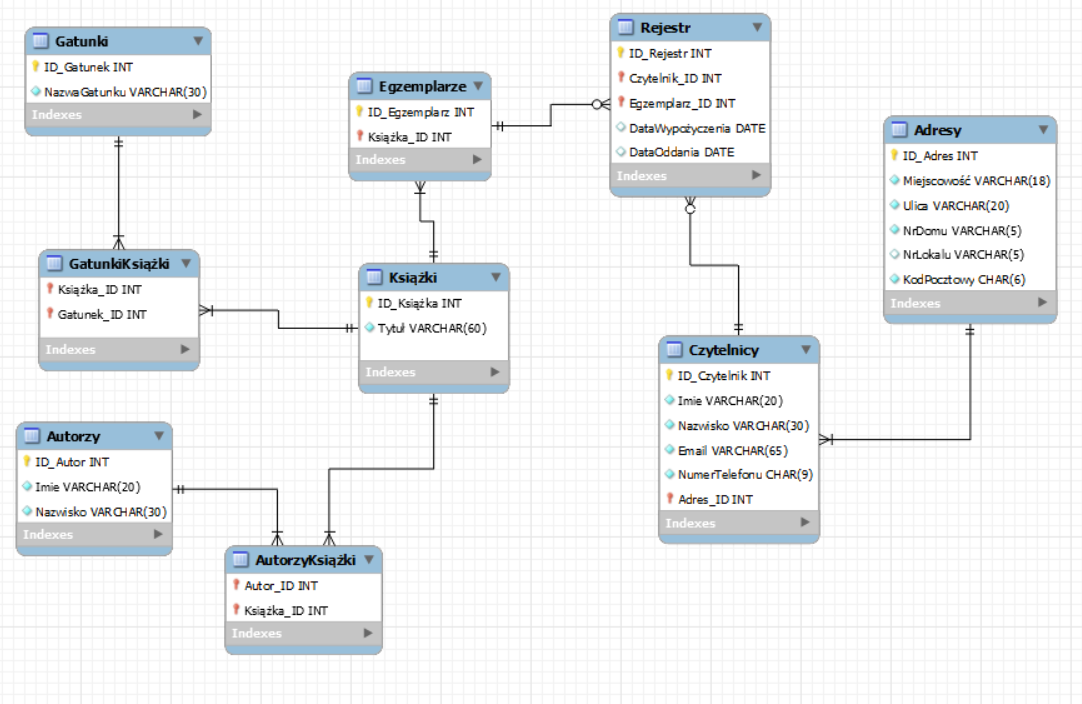
## Wyniki etapu 3

****

## Wyniki etapu 5, 7



## Wyniki etapu 9,10



## Wyświetlanie

Tworzymy perspektywe w której mamy połączone AutorKsiążki i GatunkiKsiążki

CREATE VIEW AutorzyKsiążki\_GatunkiKsiążki AS SELECT ak.Książka\_ID, Gatunek\_ID, Autor\_ID FROM AutorzyKsiążki AS ak, GatunkiKsiążki AS gk WHERE gk.Książka\_ID = ak.Książka\_ID;

Następnie dołączamy tabele Książki, Gatunki, Autorzy, w ten sposób czytelnik może przeglądać książki dostępne w bibliotece

SELECT Tytuł, Imie, Nazwisko, NazwaGatunku as Gatunek FROM AutorzyKsiążki\_GatunkiKsiążki AS ak\_gk, Książki, Autorzy, Gatunki WHERE Książki.ID\_Książka = ak\_gk.Książka\_ID AND Gatunki.ID\_Gatunek = ak\_gk.Gatunek\_ID AND Autorzy.ID\_Autor = ak\_gk.Autor\_ID;

Liczba dostępnych egzemplarzy dla danych książek

SELECT Tytuł, COUNT(ID\_Egzemplarz) AS 'Liczba dostępnych egzemplarzy' FROM Książki JOIN Egzemplarze ON Egzemplarze.Książka\_ID = Książki.ID\_Książka WHERE (ID\_Egzemplarz NOT IN (SELECT ID\_Egzemplarz FROM Egzemplarze CROSS JOIN Rejestr ON Egzemplarze.ID\_Egzemplarz = Rejestr.Egzemplarz\_ID WHERE DataOddania IS NOT NULL)) GROUP BY Tytuł ORDER BY COUNT(ID\_Egzemplarz) ASC;

Możliwość rezerwacji (bliotekarz wypełnia rejestry, jeśli nie ma egzemplarzy danej książki to ją tylko rezerwuje)

Wypożyczenie książki

INSERT INTO Rejestr VALUES (201,50, 6, '2026-01-17', '2026-01-31');

Rezerwacja książki

INSERT INTO Rejestr VALUES (202,49, 6);

Ograniczenia

INSERT INTO Rejestr VALUES (200,50, 6, '2024-01-17', '2024-01-31');

ERROR 1644 (45000): Ta książka jest aktualnie w posiadaniu kogoś innego

DELIMITER //

CREATE TRIGGER before\_insert\_Rejestr

BEFORE INSERT ON Rejestr

FOR EACH ROW

BEGIN

IF (NEW.DataWypożyczenia IS NOT NULL AND NEW.DataOddania IS NOT NULL) THEN

IF EXISTS (

SELECT 1

FROM Rejestr

WHERE Egzemplarz\_ID = NEW.Egzemplarz\_ID

AND DataOddania > NEW.DataWypożyczenia

) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Ta książka jest aktualnie w posiadaniu kogoś innego';

END IF;

END IF;

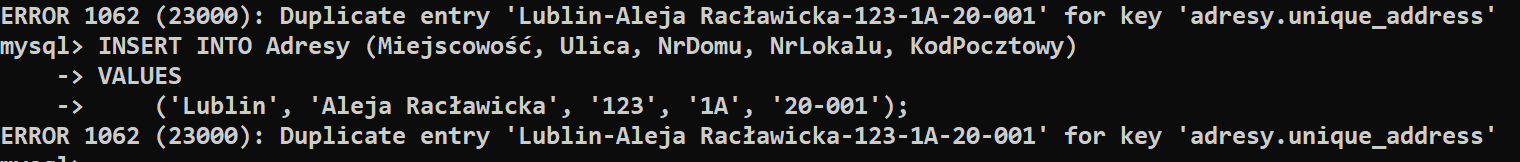
END;

//

DELIMITER ;

Mamy dla każdej tabelki zabezpieczenie przed duplikatami danych.

Przykład dla Adresy:



Zapytania istotne z punktu widzenia specyfikacji danej bazy

Bibliotekarz najpierw wyszukuje w Gatunki ID\_Gatunek, którego szuka czytelnik

SELECT ID\_Gatunek, Gatunek FROM Gatunki;

Następnie wyszukuje książki danego gatunku po Gatunek\_ID

DELIMITER // CREATE PROCEDURE KsiążkiPoGatunku(IN parametr\_gatunku INT) BEGIN SELECT Tytuł, Imie, Nazwisko, NazwaGatunku as Gatunek FROM AutorzyKsiążki\_GatunkiKsiążki AS ak\_gk JOIN Książki ON Książki.ID\_Książka = ak\_gk.Książka\_ID JOIN Autorzy ON Autorzy.ID\_Autor = ak\_gk.Autor\_ID JOIN Gatunki ON Gatunki.ID\_Gatunek = ak\_gk.Gatunek\_ID WHERE Gatunek\_ID = parametr\_gatunku; END // DELIMITER; CALL KsiążkiPoGatunku(7);

Bibliotekarz najpierw wyszukuje w Autorzy ID\_Autor, którego szuka czytelnik

SELECT ID\_Autor, Imie, Nazwisko FROM Autorzy;

Następnie wyszukuje książki danego autora po Autor\_ID

DELIMITER // CREATE PROCEDURE KsiążkiPoAutorze(IN parametr\_autor INT) BEGIN SELECT Tytuł, Imie, Nazwisko, NazwaGatunku as Gatunek FROM AutorzyKsiążki\_GatunkiKsiążki AS ak\_gk JOIN Książki ON Książki.ID\_Książka = ak\_gk.Książka\_ID JOIN Autorzy ON Autorzy.ID\_Autor = ak\_gk.Autor\_ID JOIN Gatunki ON Gatunki.ID\_Gatunek = ak\_gk.Gatunek\_ID WHERE Autor\_ID = parametr\_autor; END // DELIMITER ; CALL KsiążkiPoAutorze(7);

Najpierw sprawdza w czytelnikach z kim ma do czynienia

SELECT ID\_Czytlenik, Imie, Nazwisko FROM Czytelnicy;

Następnie wyszukuje liste książek wypożyczonych przez danego użytkownika po Czytelnik\_ID

DELIMITER // CREATE PROCEDURE WypożyczoneEgzemplarzeUżytkownika(IN parametr\_id INT) BEGIN SELECT Czytelnik\_ID, Egzemplarz\_ID AS 'Nr. wypożyczonego egzemplarza',DataWypożyczenia,DataOddania FROM Egzemplarze LEFT JOIN Rejestr ON Egzemplarze.ID\_Egzemplarz = Rejestr.Egzemplarz\_ID WHERE DataOddania IS NOT NULL AND Czytelnik\_ID = parametr\_id; END // DELIMITER ; CALL WypożyczoneEgzemplarzeUżytkownika(7);

Wyświetlanie wszystkich dostępnych egzemplarzy

SELECT ID\_Egzemplarz AS 'Nr. wypożyczonego egzemplarza', Tytuł FROM Książki JOIN Egzemplarze ON Książki.ID\_Książka = Egzemplarze.Książka\_ID WHERE (ID\_Egzemplarz NOT IN (SELECT ID\_Egzemplarz FROM Egzemplarze CROSS JOIN Rejestr ON Egzemplarze.ID\_Egzemplarz = Rejestr.Egzemplarz\_ID WHERE DataOddania IS NOT NULL));

Wszytskie wypożyczenia i rezerwacje w Rejestrze

SELECT Imie, Nazwisko, Tytuł, Egzemplarz\_ID, DataWypożyczenia, DataOddania FROM (SELECT \\* FROM Książki, Egzemplarze, Rejestr, Czytelnicy WHERE Egzemplarze.Książka\_ID = Książki.ID\_Książka AND Egzemplarze.ID\_Egzemplarz = Rejestr.Egzemplarz\_ID AND Rejestr.Czytelnik\_ID = Czytelnicy.ID\_Czytelnik) AS rkc;

Książki, które muszą być oddane po jakiejś dacie

SELECT e.ID\_Egzemplarz AS 'Numer egzemplarza', k.Tytuł, MAX(r.DataOddania) AS 'DataOddania'

FROM Egzemplarze e

JOIN Książki k ON e.Książka\_ID = k.ID\_Książka

JOIN Rejestr r ON e.ID\_Egzemplarz = r.Egzemplarz\_ID

WHERE e.ID\_Egzemplarz IN (

SELECT Egzemplarz\_ID

FROM Rejestr

WHERE DataOddania > '2024-01-29'

)

GROUP BY e.ID\_Egzemplarz, k.Tytuł

ORDER BY MAX(r.DataOddania);