

Produkt(producent, model, typ)  
PC(model, szybkość, ram, hd, cd, cena)  
Laptop(model, szybkość, ram, hd, ekran, cena)  
Drukarka(model, kolor, tym, cena)

#### 5.3.1

- b) SELECT \* FROM Drukarka WHERE EXISTS (SELECT \* FROM Drukarka ORDER BY cena DESC);
- c) SELECT \* FROM Laptop WHERE szybkość < ANY (SELECT szybkość FROM PC);
- d) SELECT model FROM PC, Laptop, Drukarka WHERE cena > ANY ((SELECT cena FROM PC) AND (SELECT cena FROM Laptop) AND (SELECT cena FROM Drukarka));
- e) SELECT model FROM Drukarka WHERE cena < ALL (SELECT cena FROM Drukarka AND kolor = „kolorowa”);
- f.)SELECT model FROM PC WHERE szybkość > ALL (SELECT szybkość FROM PC WHERE ram < ALL (SELECT ram FROM PC));

#### 5.3.2

Klasy (klasa, typ, kraj, liczbaDział, dział, wyporność)  
Okręt (nazwa, klasa, wodowanie)  
Bitwa (nazwa, data)  
Rezultat (okręt, bitwa, wynik)

- a) SELECT kraj FROM Klasy WHERE liczbaDział > ANY (SELECT liczbaDział FROM Klasy);
- b) SELECT klasa FROM Okręt WHERE nazwa EXISTS (SELECT okręt FROM Rezultat WHERE wynik < „przegrana” );
- c) SELECT nazwa FROM Okręt WHERE klasa = ALL (SELECT klasa FROM Klasy WHERE dział = „16-calowe”);
- d) SELECT bitwa FROM Rezultat WHERE okręt = ALL (SELECT nazwa FROM Okręt WHERE klasa = ALL (SELECT klasa FROM Klasy WHERE klasa = „Kongo”));
- e) SELECT nazwa FROM Okręt WHERE klasa > ANY (SELECT klasa FROM Klasy WHERE dział = „kaliber”)

#### 5.3.3

SELECT tytuł FROM Film AS Stary WHERE MIN(rok) AND tytuł = Stary.tytuł

1. Do rys. 9.1.15:

- Jaka występuje relacja między encją Granie oraz Mecz?

Związek jednoznaczny.

- Jaka relacja występuje między Mecz a Gracz?

Relacja jeden do wielu.

- Jeżeli zajdzie potrzeba przechowywania danych trenera, w której encji należy umieścić odpowiednie dane?

W Encji Drużyna.

- Gdzie można umieścić dane o stadionie, na którym odbywa się mecz?

W Encji Miejscowość. W danej miejscowości zazwyczaj jest jeden stadion.

- zaproponuj postać kwerendy SQL Create Table do utworzenia tabeli Mecz, zwróć uwagę na typy danych.

```
CREATE TABLE Mecz ( Id_Meczu INTEGER PRIMARY KEY AUTOINCREMENT, Data_rozegrania DATE,
FOREIGN KEY (Druzyna) REFERENCES Druzyna_Gosci(Id_druzyny), FOREIGN KEY (Druzyna) REFERENCES
Druzyna_Gospodarzy(Id_druzyny), Wynik_goscia INT, Wynik_gospodarza INT, FOREIGN KEY
(Miejscowosc) REFERENCES Id_Miejscowosci(Id_Miejscowosci))
```

2. Do rys. 9.1.18:

- Wskaż relacje typu wiele-do-wielu;

Czytelnik – Egzemplarz.

- Zaproponuj kwerendę SQL Create Table do utworzenia encji Książka, uwzględnij typy danych oraz ograniczenia;

```
CREATE TABLE Ksiazka(
    Id_Ksiazki uniqueidentifier NOT NULL PRIMARY KEY,
    Dziedzina uniqueidentifier NOT NULL FOREIGN KEY REFERENCES Dziedzina(Id_Dziedziny),
    Wydawnictwo uniqueidentifier NOT NULL FOREIGN KEY REFERENCES Wydawnictwo(Id_Wydawnictwa),
    Jezyk uniqueidentifier NOT NULL FOREIGN KEY REFERENCES jezyk(Id_Jezyk),
    Tytul varchar(255) NOT NULL,
    Tytul_Oryginalu varchar(255),
    ISBN numeric(13,0),
    Rok_Wydania numeric(4,0),
    Numer_Wydania varchar(30),
    Streszczenie text,
);
```

- Zbuduj kwerendę SQL wybierającą, która odpowiada na pytanie: 'Podaj tytuły książek, które wypożyczył i nie zwrócił czytelnik o nazwisku Kowalski'

Brakowało mi klucza obcego do tabeli książka w tabeli egzemplarz.

Poniższe zapytanie uwzględnia klucz obcy Id\_Ksiazki w tabeli egzemplarz wskazujący pole Id\_ksiazki w tabeli Ksiazka

```
SELECT k.tytul FROM ksiazka k
INNER JOIN egzemplarz e ON k.Id_Ksiazki = e.Id_ksiazki
WHERE EXISTS
```

```
(  
SELECT 1 FROM wypozyzenie w  
INNER JOIN czytelnik c ON w.numer_czytelnika = c.numer_czytelnika  
WHERE  
e.Sygnatura = w.sygnatura  
AND  
c.Nazwisko = 'Kowalski'  
AND  
w.Data_Zwrotu < GETDATE()  
)
```

- Przypuśćmy, że książka uległa zniszczeniu, uszkodzeniu i nie może być więcej używana. Nie chcemy natomiast tracić historii jej wypożyczeń. Zaproponuj rozwiązanie zmian w schemacie uwzględniającą daną sytuację.

Dodanie boolowskiego pola które określa czy książka jest zniszczona 1- tak, 0- nie

```
ALTER TABLE egzemplarz  
ADD Czy_Uszkodzona BIT
```