



Bazy danych SQL

Marcin Erbel



- Kim jestem?
- Poznajmy się!
- Jakie są Wasze oczekiwania?
- Czego chcecie się nauczyć?
- Jak będą wyglądały nasze spotkania?
- Wasze sugestie?

Agenda



- Podstawowe pojęcia
- SQL
- Typy danych
- Operacje
- Relacje
- Projektowanie bazy danych
- Diagramy ERD
 - Triggery
 - Indeksy
 - Widoki
- Procedury składowane
 - Transakcje



Podstawowe pojęcia

Baza danych

- Zbiór danych zapisanych zgodnie z określonymi regułami.



Schemat bazy danych

- Zawiera tabele i określa relacje między nimi.
- Przedstawia strukturę bazy danych.



SZBD

- SZBD – System Zarządzania Bazą Danych (ang. Database Management System).
- Klient – Serwer: DB2, MySQL, MariaDB, Oracle, PostgreSQL.
- Standalone: Access, Kexi.



RDBMS

- System zarządzania relacyjną bazą danych
(ang. Relational Database Management System RDBMS).



Rodzaje baz danych

- Bazy proste
- Bazy złożone



Rodzaje baz danych prostych

- Kartotekowe
- Hierarchiczne



Rodzaje baz danych złożonych

- Relacyjne
- Obiektowe
- Relacyjno-obiektowe
- Temporalne
- Nierelacyjne (NoSQL)



Relacyjny model danych

- Dane przedstawione w postaci relacyjnej.
- Reprezentacja relacji ma miejsce w tablicach.
- Relacje - zbiór rekordów o identycznej strukturze.
- Schematy baz danych – zgrupowane relacje.



CRUD

- CREATE
- READ
- UPDATE
- DELETE



CRUD

Działanie	Instrukcja SQL	HTTP	DDS
CREATE	INSERT	PUT/POST	WRITE
READ/RETRIEVE	SELECT	GET	READ/TAKE
UPDATE	UPDATE	POST/PUT/PATCH	WRITE
DELETE (DESTROY)	DELETE	DELETE	DISPOSE



MySQL - instalacja

- MySql Server
- MySql Workbench

<https://www.mysql.com/downloads>



MySQL - konfiguracja

- Tworzymy schemat bazy danych
- Tworzymy użytkownika
- Cieszymy się działającą bazą!



MySQL – dane testowe

- Pobieramy przykładową bazę danych:
https://github.com/datacharmer/test_db

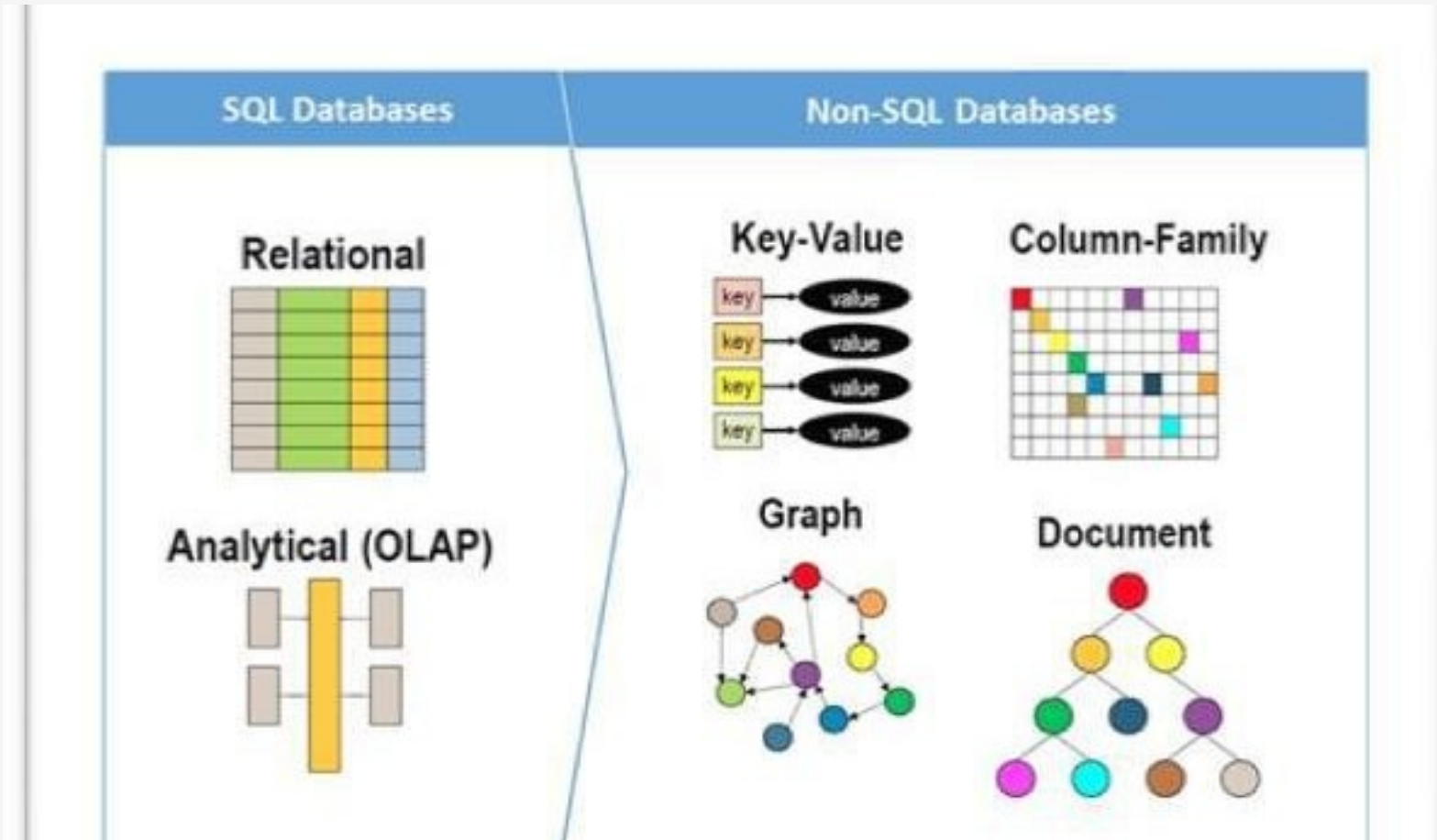




SQL



- Wieloplatformowy
- Statycznie i silnie typowany
- Wieloparadygmatowy
- Posiada wiele implementacji



Składnia SQL

- **SQL DML** (ang. *Data Manipulation Language* – „język manipulacji danymi”),
- **SQL DDL** (ang. *Data Definition Language* – „język definicji danych”),
- **SQL DCL** (ang. *Data Control Language* – „język kontroli nad danymi”).
- **SQL DQL** (ang. *Data Query Language* – „język definiowania zapytań”).



SQL DML

- **INSERT** INTO (...) VALUES (...);
- **UPDATE** (...) SET (...) WHERE (...);
- **DELETE** FROM (...) WHERE (...);



SQL DDL

- **CREATE** DATABASE (...)
- **DROP** DATABASE (...)
- **ALTER** TABLE (...) ADD (...)



SQL DCL

- **GRANT** PRIVILAGE ON (...) TO (...)
- **REVOKE** PRIVILAGE ON (...) FROM (...)



SQL DQL

- **SELECT** (...) FROM (...) WHERE (...)



WHERE

OPERATOR	OPIS
=	RÓWNY
<> / !=	RÓŻNY
>	WIĘKSZY
<	MNIEJSZY
>=	WIĘKSZY BĄDŹ RÓWNY
<=	MNIEJSZY BĄDŹ RÓWNY



AND / OR / NOT

AND	1	0	?
1	1	0	?
0	0	0	0
?	?	0	?

OR	1	0	?
1	1	1	1
0	1	0	?
?	1	?	?

CONDITION	NOT
1	0
0	1
?	?



SELECT DISTINCT

- SELECT **DISTINCT** FROM TABLE
- Usuwa powtarzające się wiersze przed zwróceniem ich na zewnątrz.
- Wiersze uznawane są za takie same tylko wtedy jeżeli wszystkie wartości ich komórek w odpowiednich kolumnach są sobie równe.



ORDER BY

- `SELECT (...) FROM (...) ORDER BY [ASC|DESC]`
- ASC – (ang. ascending) – wstępująco – wartość domyślna
- DESC – (ang. descending) – zstępująco



NULL ?

- SELECT (...) FROM (...) WHERE (...) IS NULL
- SELECT (...) FROM (...) WHERE (...) IS NOT NULL



SELECT TOP

- ORACLE DB
 - `SELECT (...) FROM (...) WHERE (...) ROWNUM <= (...)`
- MYSQL:
 - `SELECT (...) FROM (...) WHERE (...) LIMIT (...)`
- SQL SERVER:
 - `SELECT TOP (...) FROM (...) WHERE (...)`



MIN / MAX

- SELECT **MIN**(...) FROM (...) WHERE (...)
- SELECT **MAX**(...) FROM (...) WHERE (...)



COUNT / AVG / SUM

- SELECT **COUNT**(...) FROM (...) WHERE (...)
- SELECT **AVG**(...) FROM (...) WHERE (...)
- SELECT **SUM**(...) FROM (...) WHERE (...)



LIKE

- SELECT (...) FROM (...) WHERE (...) LIKE (...)
- %- znak we wzorcu reprezentujący 0,1,wiele znaków
- _- znak we wzorcu reprezentujący pojedynczy znak



IN

- SELECT (...) FROM (...) WHERE (...) IN (...)
- SELECT (...) FROM (...) WHERE (...) IN (SELECT ...)



ALIAS

- SELECT (...) **AS** (...) FROM (...)



GROUP BY

- `SELECT SUM (...) FROM (...) GROUP BY (...)`

Operator pozwala na grupowanie wyników zapytania, co jest wykorzystywane w szczególności przy zastosowaniu funkcji agregujących.





Visual Explanation of SQL Join Clauses

Common Join Clauses

Inner Join

Return rows where key values intersect



```
SELECT A.col, B.col  
FROM A  
INNER JOIN B  
ON A.Key = B.Key
```

Left Outer Join

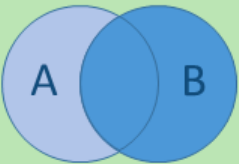
Return all rows from A, match B, when no match found, return NULLS for columns in B.



```
SELECT c1, c2  
FROM A  
LEFT OUTER JOIN B  
ON JOIN A.Key = B.Key
```

Right Outer Join

Return all rows from B, match A, when no match found, return NULLS for columns in A.

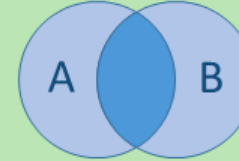


```
SELECT c1, c2  
FROM A  
RIGHT OUTER JOIN B  
ON JOIN A.Key = B.Key
```

Exotic Join Clauses

Full Join

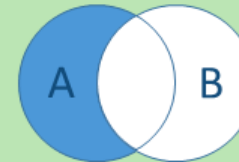
Left and Right Join Combined – Return all rows that Match A and B, when no match, found, return nulls for that table



```
SELECT c1, c2  
FROM A  
FULL OUTER JOIN B  
ON JOIN A.Key = B.Key
```

Minus (A - B)

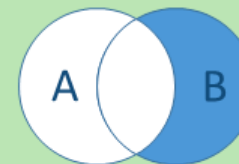
Return row from A only if key value is exclusive to A.



```
SELECT c1, c2  
FROM A  
LEFT OUTER JOIN B  
ON A.Key = B.Key  
WHERE B.Key is NULL
```

Minus (B - A)

Return row from A only if key value is exclusive to B.



```
SELECT c1, c2  
FROM A  
RIGHT OUTER JOIN B  
ON JOIN A.Key = B.Key  
WHERE A.Key is NULL
```



Typy danych

Typy liczbowe

- TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT
- FLOAT
- DOUBLE
- DECIMAL



Typy daty

- DATE
- DATETIME
- TIMESTAMP
- TIME
- YEAR



Typy znakowe

- CHAR
- VARCHAR



Typy binarne

- TINYBLOB, TINYTEXT
- BLOB, TEXT
- MEDIUMBLOB, MEDIUMTEXT
- LONGBLOB, LONGTEXT



Typy łańcuchowe

- ENUM
- SET





Operacje



Operacje na datach i czasie – klauzula INTERVAL

Rodzaj interwału	Jednostka
MICROSECOND	mikrosekundy
SECOND	sekundy
MINUTE	minuty
HOURL	godziny
DAY	dni
WEEK	tygodnie
MONTH	miesiące

```
SELECT DATE_ADD('data', INTERVAL 1 DAY)
```

Operacje dla znalezienia różnicy pomiędzy datami

- DATEDIFF('data', 'data')
- TIMEDIFF('czas', 'czas')





Formatowanie daty i czasu

%a	Skrócona nazwa dnia tygodnia (Sun..Sat / niedz..sob)
%b	Skrócona nazwa miesiące (Jan..Dec / sty..gru)
%c	Miesiąc, liczbowo (0..12)
%D	Dzień miesiąca z angielskim sufiksem (0th, 1st, 2nd, 3rd, ...)
%d	Dzień miesiąca, liczbowo (00..31)
%e	Dzień miesiąca, liczbowo (0..31)
%f	Mikrosekundy (000000..999999)
%H	Godzina (00..23)
%h	Godzina (01..12)
%I	Godzina (01..12)
%i	Minuty, liczbowo (00..59)
%j	Dzień w roku (001..366)
%k	Godzina (0..23)
%l	Godzina (1..12)



Relacje

Rodzaje relacji

Zasadniczo wyróżniamy trzy typy relacji:

- 1:1
- 1:N
- N:M



WIĘZY INTEGRALNOŚCI

- PRIMARY KEY – unikalność wartości w kolumnie. W tabeli może być tylko jeden klucz główny. Zapewnia nie występowanie wartości NULL.
- UNIQUE – unikalność wartości w kolumnie. Może być więcej niż jeden. Umożliwia występowanie wartości NULL.
- NOY NULL – zapobiega ustawianiu wartości NULL.
- CHECK – wartość musi spełniać warunek logiczny.



WIĘZY INTEGRALNOŚCI

- FOREIGN KEY – klucz obcy wykorzystywany przy tworzeniu relacji. Zapewnia, że rekord w tabeli podrzędnej zawsze będzie miał swojego odpowiednika w tabeli nadrzędnej.





WIĘZY INTEGRALNOŚCI

- **ON DELETE** – akcja wykonywana przy usuwaniu danego wiersza.
- **ON UPDATE** – akcja wykonywana przy modyfikacji danego wiersza.
- **NO ACTION** - wyłącza mechanizm klucza obcego dla danej operacji.
- **RESTRICT** - nie pozwala na dokonanie zmian naruszających powiązanie.
- **CASCADE** - nakazuje zmianom na propagację kaskadową wzdłuż drzewa powiązanych tabel.
- **SET NULL** - ustawia odpowiednie atrybuty powiązanych tabel, dotąd wskazujące na usuwany/modyfikowany element klucza obcego, na wartość NULL, jeśli definicja tabeli to dopuszcza.
- **SET DEFAULT** - ustawia odpowiednie atrybuty powiązanych tabel, dotąd wskazujące na usuwany/modyfikowany element klucza obcego, na wartość DEFAULT, jeśli definicja tabeli to dopuszcza.



Diagramy ERD

Diagramy ERD

- Diagramy Związków Encji (ang. Entity-Relationship Diagram) – graficzny sposób przedstawienia związków pomiędzy encjami.
- Notacja Barkera.
- Notacja Martina (notacja kruczej stopki).



ERD

- Encja – istniejący, rozróżnialny obiekt, np. kurs, wykładowca, student, przedmiot, szkoła.
- Atrybut – informacja charakteryzująca encję.
- Związek/Relacja – powiązanie pomiędzy encjami składające się z nazwy, krotności oraz opcjonalności.



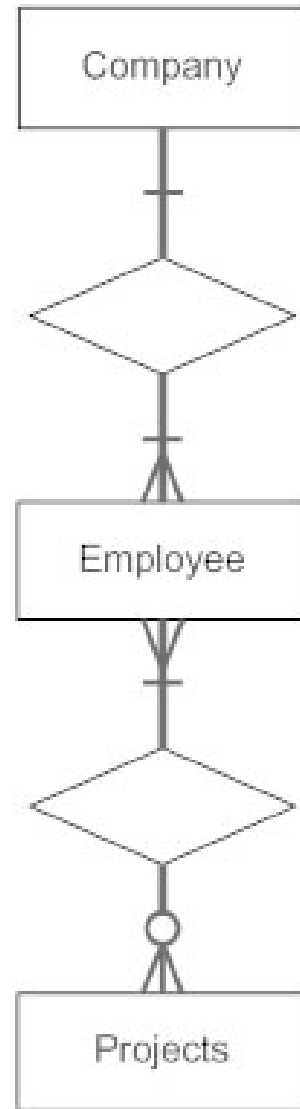
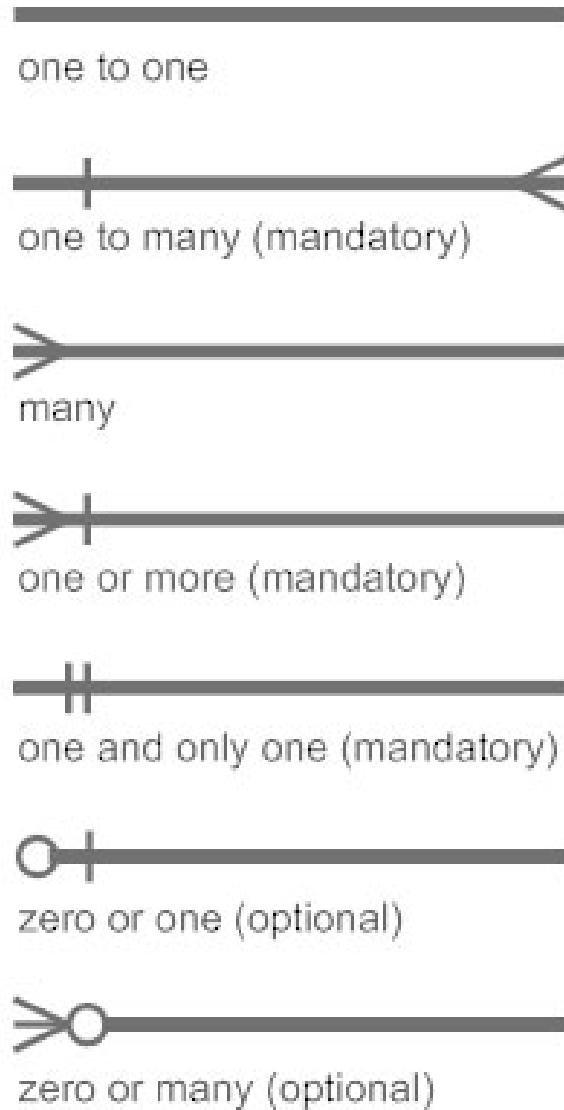
Logiczne związki pomiędzy encjami

- Opcjonalność
- Krotność
 - 1:1
 - 1:N
 - M:N



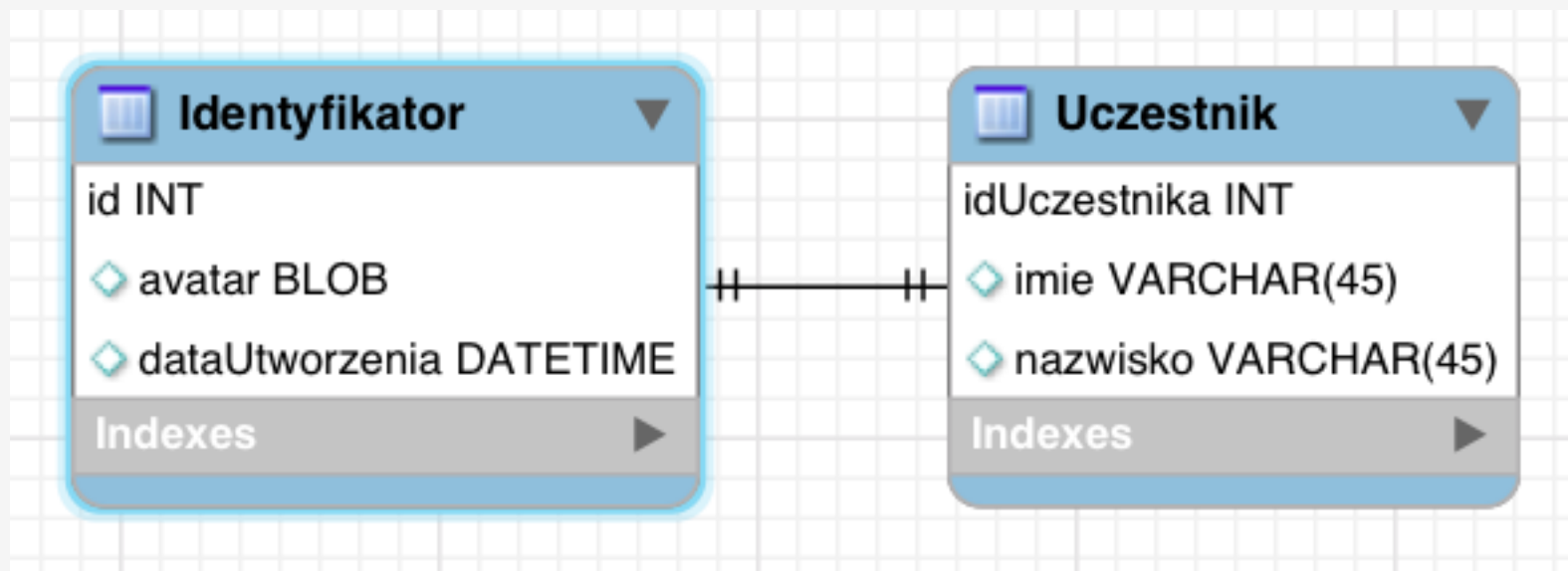


ERD - Relacje

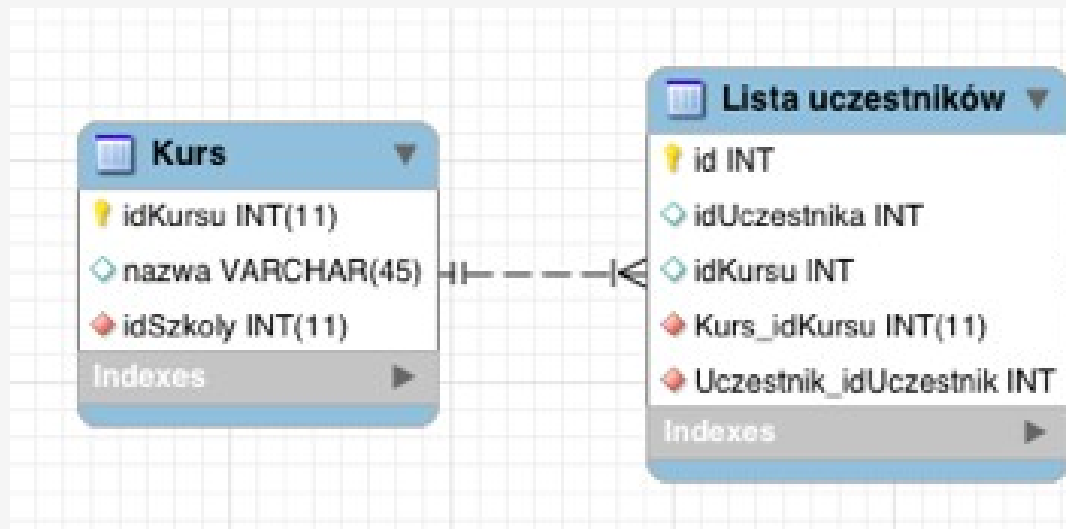


Źródło: <https://www.smartdraw.com/entity-relationship-diagram>

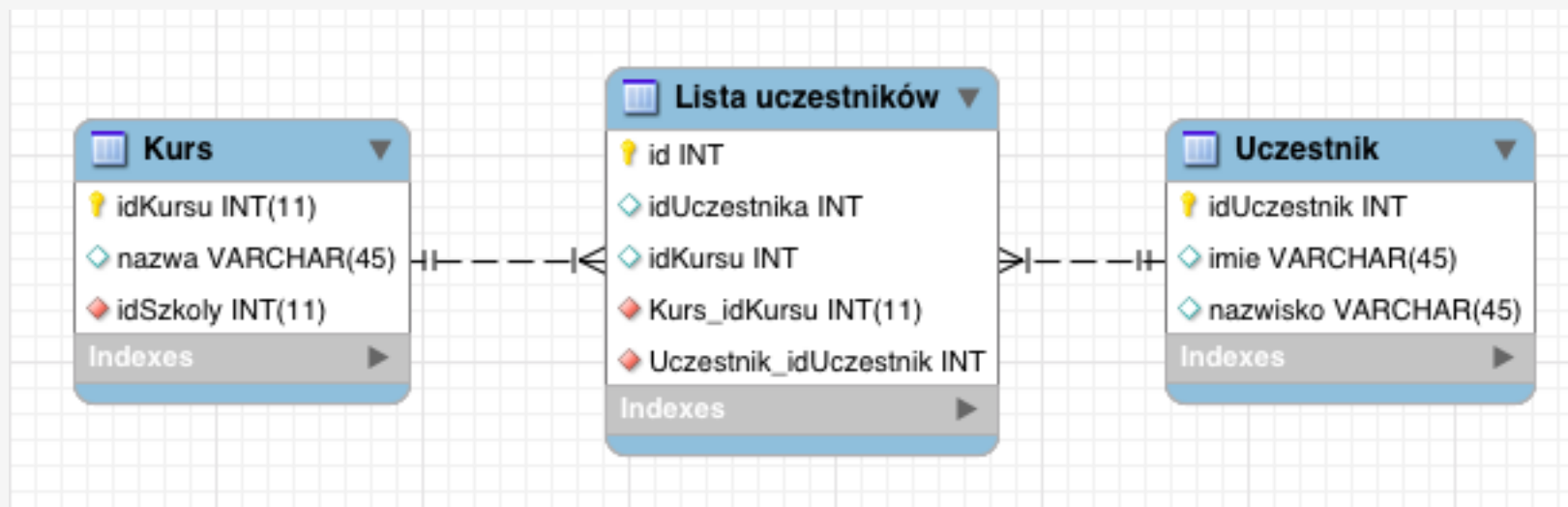
Relacja jeden do jeden



Relacja jeden do wielu



Relacja wiele do wielu





Projektowanie bazy danych

Spróbujmy zamodelować rzeczywistość

- Przykłady schematów baz danych
- Spróbujmy zamodelować nasz kurs





CREATE TABLES

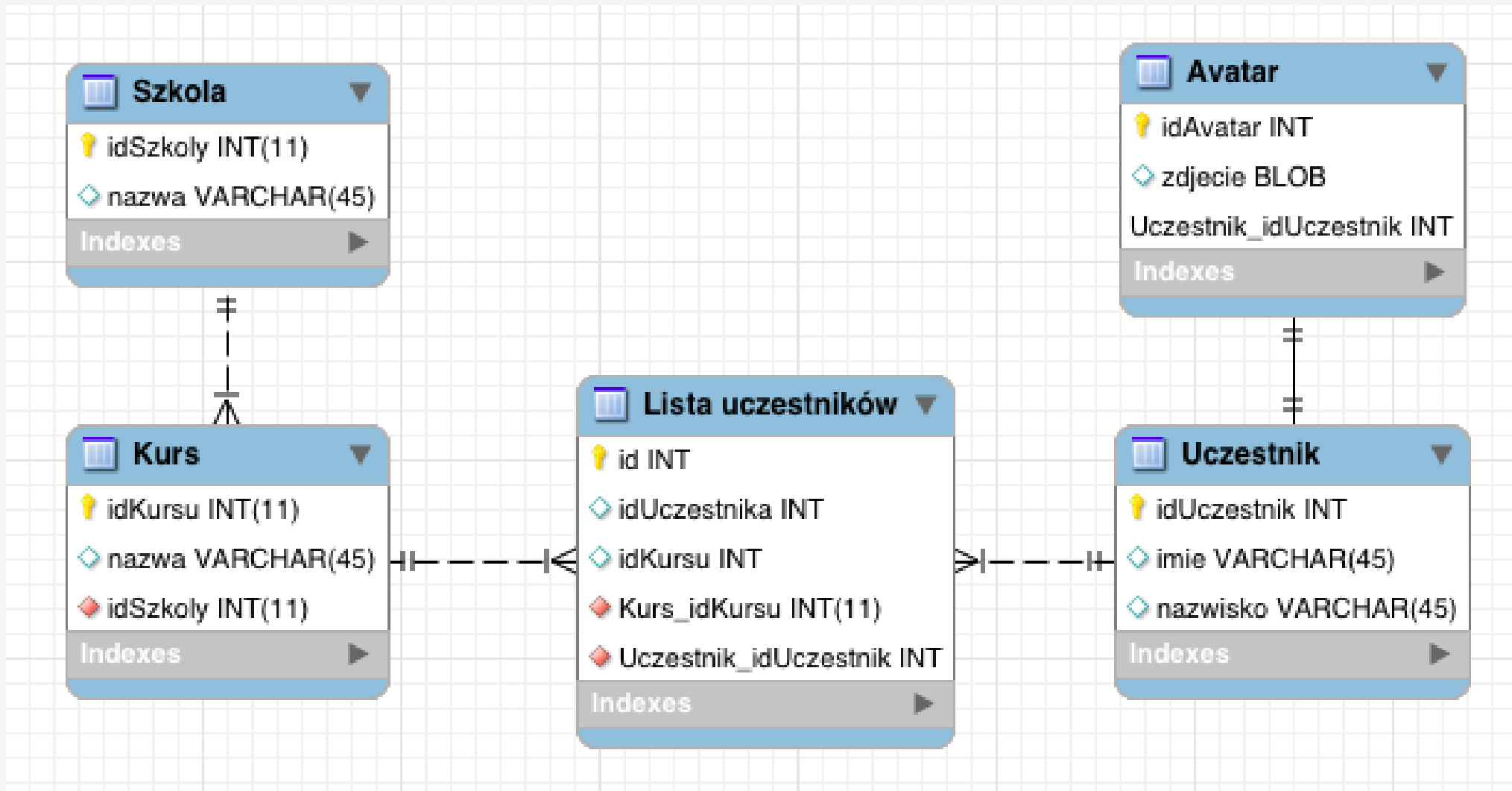
```
CREATE TABLE IF NOT EXISTS `SDA`.`Szkola` (  
  `idSzkoły` INT NOT NULL,  
  `nazwa` VARCHAR(45) NULL,  
  PRIMARY KEY (`idSzkoły`))
```

- Jaki to rodzaj relacji?
- Jakie tabele możemy jeszcze stworzyć?
- Jakich relacji będziemy musieli użyć?

```
CREATE TABLE IF NOT EXISTS `SDA`.`Kurs` (  
  `idKursu` INT NOT NULL,  
  `nazwa` VARCHAR(45) NULL,  
  `idSzkoły` INT NOT NULL,  
  PRIMARY KEY (`idKursu`),  
  INDEX `idSzkoły_idx` (`idSzkoły` ASC),  
  CONSTRAINT `idSzkoły`  
    FOREIGN KEY (`idSzkoły`)  
    REFERENCES `SDA`.`Szkola` (`idSzkoły`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)
```



Przykładowy model kursu programowania z uczestnikami





Triggery

Trigger / Wyzwalacz

- Niemożliwe jest bezpośrednie wywołanie wyzwalacza dyrektywą CALL
- Wyzwalacze to specjalne typy procedur składowanych wywoływanych wykonaniem instrukcji typu:
 - **INSERT**
 - **UPDATE**
 - **DELETE**



Zastosowanie triggerów

- Kaskadowe aktualizowanie danych w powiązanych relacjami tabelach.
- Sprawdzanie poprawności danych.
- Sprawdzanie danych zmodyfikowanych w dowolnej liczbie wierszy.
- Monitorowanie aktywności użytkowników.



Trigger

```
CREATE TRIGR (...)  
BEFORE UPDATE ON (...)  
FOR EACH ROW  
BEGIN  
...  
END;
```

```
DROP TRIGGER (...)
```





Indeksy

INDEKSY

- Używane dla danych o które często odpytujemy.
- Mogą być ustawiane na jednej albo wielu kolumnach.
- Działają jak wskaźnik do wierszy w tabeli.
- Tworzenie indeksu przez instrukcję INDEX
- INDEX(BLOB_COL(10)) – utworzy indeks pierwszych 10 znaków kolumny





INDEKSY +/-

ZALETY

- Przyspieszają wyszukiwanie
- Wymuszają unikatowość wierszy

WADY

- Zabierają miejsce na dysku
- Zwiększają obciążenie systemu



Co warto indeksować?

WARTO

- Klucze podstawowe i obce.
- Pola, po których często szukamy.
- Pola, do których dostęp następuje w określonej kolejności.

NIE WARTO

- Pola rzadko używane w zapytaniach.
- Pola nie zawierające wartości unikatowych
- Pola zawierające dane typu BLOB/IMAGE



Widoki

Widoki

- Używane do wyświetlania konkretnych kolumn z tabeli.
- Wywoływanie widoku odbywa się za pomocą instrukcji SELECT, gdzie po dyrektywie FROM podawana jest nazwa widoku.



WIDOK

CREATE OR REPLACE VIEW (...) AS
SELECT (...) FROM (...)

DROP VIEW





Procedury składowane

Procedura składowana

- Cel procedury to agregacja zapytań i przechowywanie ich na serwerze.
- Praca z procedurami składowanymi jest łatwiejsza, wygodniejsza, bezpieczniejsza.
- Procedury są osadzone na serwerze bazodanowym i przenoszone wraz z backupem.



Procedura składowana

CREATE PROCEDURE (...)

(IN (...) varchar (64))

UPDATE (...)

SET (...) = (...)

WHERE (...);

DROP PROCEDURE (...)



SQL Injection

- Metoda ataku komputerowego polegająca na niedostatecznym typowaniu danych użytkownika, które są później wykorzystywane przy wykonaniu zapytań do bazy danych.
- Przy używaniu procedur składowanych aplikacja nie ma bezpośredniego wpływu na kształt zapytania.

Źródło: https://pl.wikipedia.org/wiki/SQL_injection



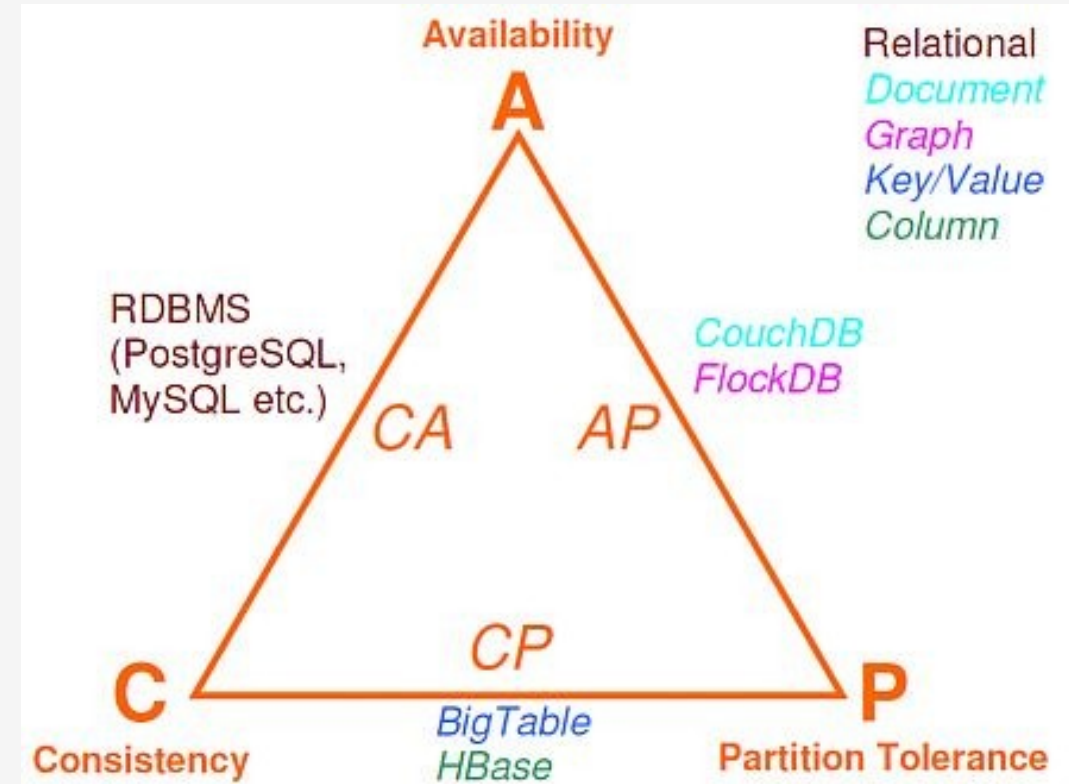


Transakcije



ACID & CAP

- Zbiór właściwości gwarantujących poprawne przetwarzanie transakcji:
- Niepodzielność (ang. atomicity)
- Spójność (ang. consistency)
- Izolacja (ang. isolation)
- Trwałość (ang. durability)



Transakcja

- Współbieżny dostęp do zawartości bazy danych dostarczając niezbędnych mechanizmów synchronizacji.
- Jak działa przelew? Co się stanie jeżeli zostanie przerwany w trakcie?
- Transakcja jest ciągiem operacji do wspólnego, niepodzielnego wykonania.



Niepodzielność

- Cała transakcja albo żadna.
- Nie może zostać podzielona.
- Nie może zostać zrealizowana częściowo.



Spójność / Integralność

- Muszą zostać spełnione wszystkie warunki poprawności nałożone przez bazę danych po zaakceptowaniu transakcji.



Izolacja

- Transakcje nie mogą na siebie wpływać.
- Efekty równoległego wykonywania transakcji muszą być szeregowalne.



Trwałość

- Efekty działania transakcji muszą przynieść stały efekt w bazie danych.



Transakcje w SQL

- BEGIN
- COMMIT;
- ROLLBACK;





Kontakt

Marcin Erbel

tel: 609 620 905

mail: marcinerbel90@gmail.com

fb: facebook.com/marcin.erbel