

BAZY DANYCH SQL: PODSTAWY SQL

Bazy danych SQL

SDA Academy - Materiały pomocnicze

Opracował : Dariusz Konciewicz

Dzień 1 - Plan

1. Wstęp - relacyjne bazy danych
2. SELECT - pobieranie danych z pojedynczej tabeli
3. DML - Wstawianie, aktualizacja i usuwanie rekordów z tablicy
4. Złączenia - pobieranie danych z wielu tabel
 - Inner join
 - Outer join
 - Self join

Dzień 2 - Plan

1. Agregowanie danych

- Funkcje agregujące dane (Min, Max, Avg, Count, ...)
- klauzule GROUP BY, HAVING

2. Funkcje wbudowane SQL

3. Widoki (Views)

4. Złożone zapytania

- Subqueries, operatory: IN, EXISTS

5. Elementy projektowania bazy danych

6. Zarządzanie bazą danych - Język DDL Data definition Language

Dzień 3 - Plan

1. Procedury składowane (Stored procedures)
2. Wyzwalacze i eventy (Triggers and Events)
3. Transakcje i współbieżność
4. Dodatek A) Indeksowanie tabel w bazie danych
5. Dodatek B) Bezpieczeństwo baz danych

Materiały do szkolenia

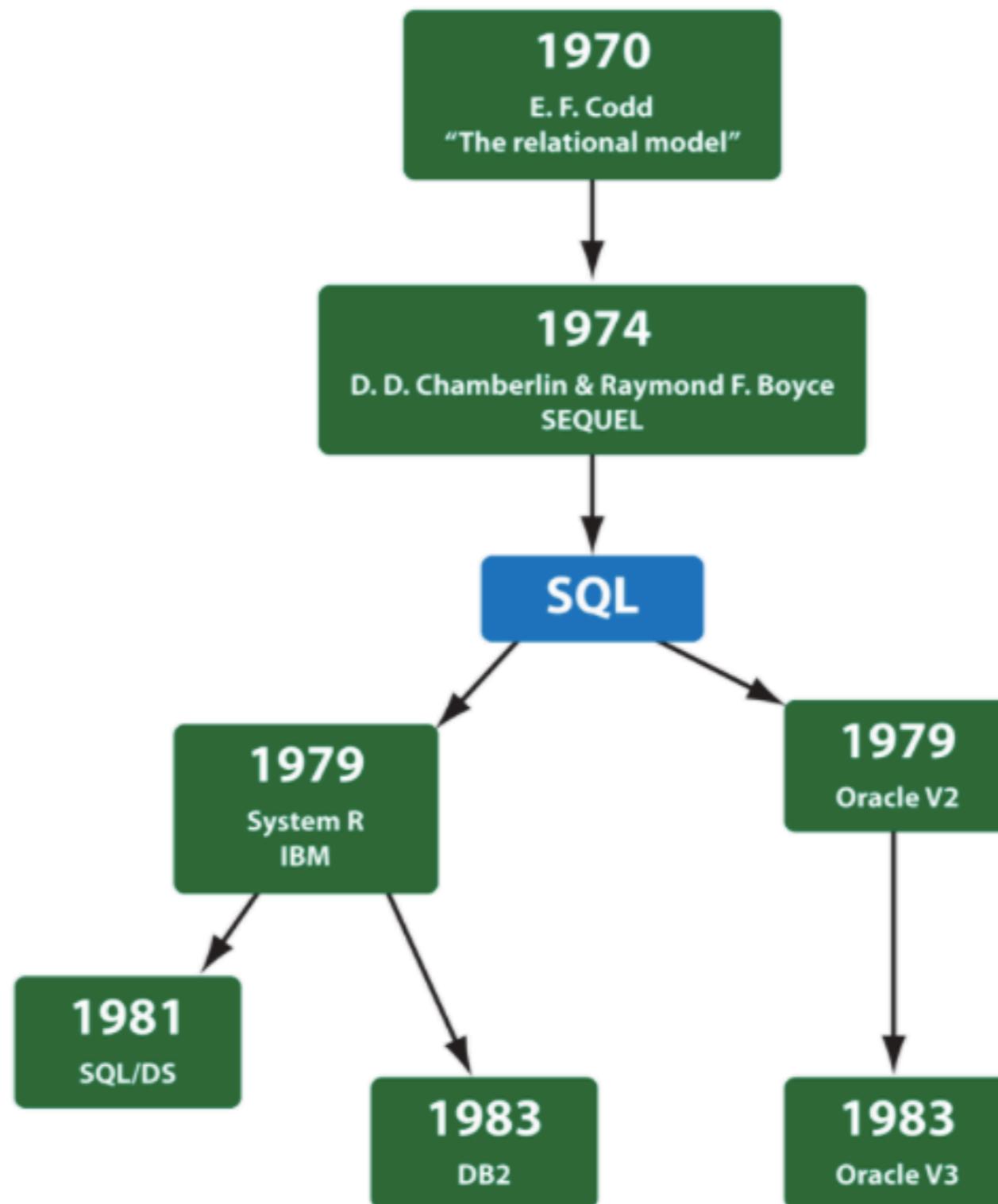
- Prezentacje (slajdy)
 - [https://java.pl.sdacademy.pro/prezentacje/bazy danych sql/](https://java.pl.sdacademy.pro/prezentacje/bazy_danych_sql/)
- E-Book
 - [https://java.pl.sdacademy.pro/slides/bazy danych sql/](https://java.pl.sdacademy.pro/slides/bazy_danych_sql/)
- Zadania
 - [https://java.pl.sdacademy.pro/slides/bazy danych sql/](https://java.pl.sdacademy.pro/slides/bazy_danych_sql/)
- Instalacja MySQL + MySQLWorkbench (instrukcja na YT)
 - <https://www.youtube.com/watch?v=UduVPtGGytY>

Dzień 1 -

Wprowadzenie

Materiały dodatkowe i Zadania

Krótką historia rozwoju S.Q.L



S.Q.L vs sequel ?



Dzień 1 - SQL - Pobieranie danych z pojedynczej tabeli

Zadania

Zadanie 1 - klauzula SELECT .. FROM

- W bazie **sql_store** stwórz zapytanie, które zwróci wszystkie produkty (z tabeli products) opisane kolumnami :
 - name,
 - unit price,
 - new price (= unit price * 1.1).

Result Grid			
	name	unit price	new price
▶	Foam Dinner Plate	1.21	1.331
	Pork - Bacon,back Peameal	4.65	5.115
	Lettuce - Romaine, Heart	3.35	3.685
	Brocolinni - Gaylan, Chinese	4.53	4.983
	Sauce - Ranch Dressing	1.63	1.793
	Petit Baguette	2.39	2.629
	Sweet Pea Sprouts	3.29	3.619
	Island Oasis - Raspberry	0.74	0.814
	Longan	2.26	2.486
	Broom - Push	1.09	1.199

Zadanie 2 - klauzula WHERE

- Znajdz zamówienia zlozone w roku 2019.

Result Grid							Filter Rows:	Search	Edit:	Export/Import:
	order_id	customer_id	order_date	status	comments	shipped_date	shipper_id			
▶	1	6	2019-01-30	1	NULL	NULL	NULL			
	NULL	NULL	NULL	NULL	NULL	NULL	NULL			

Zadanie 3 - Operator IN, NOT IN

- Odszukaj produkty spełniające warunek:
 - quantity in stock = { 49, 38, 72 }

Result Grid		Filter Rows:	Search	Edit:	Export/Import:
product_id	name	quantity_in_stock	unit_price		
2	Pork - Bacon,back Peameal	49	4.65		
3	Lettuce - Romaine, Heart	38	3.35		
NULL	NULL	NULL	NULL		

Zadanie 4 - Operator BETWEEN ... AND ...

- Odszukaj Klientów (Customers) urodzonych pomiędzy 1/1/1990 oraz 1/1/2000

Zadanie 5 - Operatory LIKE i NOT LIKE

- Odszukaj klientów (Customers) tych którzy spełniają warunki :
 1. address zawiera TRAIL lub AVENUE
 2. numer telefonu jest zakończony na 9

Result Grid									
Filter Rows: <input type="text" value="Search"/> Edit: Export/Import:									
customer_id	first_name	last_name	birth_date	phone	address	city	state	points	
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947	
9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486	
10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796	
NUL	NUL	NUL	NUL	NUL	NUL	NUL	NUL	NUL	

Zadanie 6 - Operatory IS NULL, IS NOT NULL

- Znajdź zamówienia (orders), które nie są wysłane (shipped) :

Zadanie 7 - klauzula ORDER BY

- W bazie **sql_store** stwórz zapytanie, które da rezultat jak poniżej :

Result Grid | Filter Rows: Search |

	order_id	product_id	quantity	unit_price	
▶	2	1	2	9.10	
	2	4	4	1.66	
	2	6	2	2.94	
	NULL	NULL	NULL	NULL	

Zadanie 8 - klauzula LIMIT

- Wyświetl 3 najbardziej lojalnych klientów (podpowiedź : im klient ma więcej punktów tym jest bardziej lojalny).

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
5	Clemmie	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Dzień 1 DML - Data Modification Language

Zadania

Zadanie 1- wstawianie rekordów do tablicy

- Wstaw trzy nowe wiersze opisujące produkty (products).
- Sprawdź wynik operacji.

Result Grid		Filter Rows:	Search	Edit:	Export/Import:	
product_id	name	quantity_in_stock	unit_price			
5	Sauce - Ranch Dre...	94	1.63			
6	Petit Baguette	14	2.39			
7	Sweet Pea Sprouts	98	3.29			
8	Island Oasis - Rasp...	26	0.74			
9	Longan	67	2.26			
10	Broom - Push	6	1.09			
15	Product1	10	1.95			
16	Product2	11	1.95			
17	Product3	12	1.95			

Zadanie 2 - Tworzenie kopii tablicy

- Stwórz nową tablicę invoices_archive dla rekordów posiadających określoną (niepustą) datę płatności

Dzień 1 - Pobieranie danych z wielu tabel

Zadania

Zadanie 1 - Inner Join

- Dla tablicy order_items wykonaj złączenie z tabelą products, wyświetl wynik tak, żeby po kolumnie : product_id, wyświetlała się nazwa produktu. Użyj aliasów.

The screenshot shows the MySQL Workbench interface with a result grid. The grid has columns: order_id, product_id, name, quantity, and unit_price. The data consists of 14 rows, each representing an item in an order. The 'name' column displays the product names as specified in the query. The 'Result 8' label at the bottom indicates the total number of rows returned.

order_id	product_id	name	quantity	unit_price
2	1	Foam Dinner Plate	2	9.10
6	1	Foam Dinner Plate	4	8.65
10	1	Foam Dinner Plate	10	6.01
5	2	Pork - Bacon,back Peameal	3	9.89
6	2	Pork - Bacon,back Peameal	4	3.28
3	3	Lettuce - Romaine, Heart	10	9.12
4	3	Lettuce - Romaine, Heart	7	6.99
6	3	Lettuce - Romaine, Heart	4	7.46
7	3	Lettuce - Romaine, Heart	7	9.17
1	4	Brocolinni - Gaylan, Chinese	4	3.74
2	4	Brocolinni - Gaylan, Chinese	4	1.66
6	5	Sauce - Ranch Dressing	1	3.45
8	5	Sauce - Ranch Dressing	2	6.94

Zadanie 2 - złączenie wielu tabel

- Rezultat złączenia : payments[] + clients[] + payment_methods jest skomplikowany.
- Uprość go tak aby wyświetlic jedynie : payments.date, invoice_id, payments.amount, clients.name, payment_methods.name

	date	invoice_id	amount	name	name
▶	2019-01-03	6	74.55	Vinte	Credit Card
	2019-01-11	11	0.03	Yadel	Credit Card
	2019-01-15	15	80.31	Yadel	Credit Card
	2019-01-15	17	68.10	Yadel	Credit Card
	2019-01-08	18	10.00	Topiclounge	Cash
	2019-02-12	2	8.18	Topiclounge	Credit Card
	2019-01-26	13	87.44	Topiclounge	Credit Card
	2019-01-08	18	32.77	Topiclounge	Credit Card

Zadanie 3 - OUTER JOIN

- W bazie **sql_store** zbuduj zapytanie dające następujący rezultat :
 - mamy trzy kolumny: product_id, name + quantity
 - zapytanie musi znajdowac produkty nawet jeśli liczba zamówionych produktów wynosi 0 (NULL)

product_id	name	quantity
1	Foam Dinner Plate	2
1	Foam Dinner Plate	4
1	Foam Dinner Plate	10
2	Pork - Bacon,back Peameal	4
2	Pork - Bacon,back Peameal	3
2	Pork - Bacon,back Peameal	4
3	Lettuce - Romaine, Heart	10
3	Lettuce - Romaine, Heart	7
3	Lettuce - Romaine, Heart	4
3	Lettuce - Romaine, Heart	7

Zadanie 4 - wielokrotne OUTER JOIN

- Zbuduj zapytanie Outer Join zawierające : order_date, order_id (customer).first_name, shipper oraz status (nazwa). Każde zamówienie powinno posiadać swojego klienta.

order_date	order_id	customer	shipper	status
2018-06-08	8	Clemmie	NULL	Processed
2018-11-18	6	Levy	NULL	Processed
2017-01-22	4	Ines	NULL	Processed
2017-12-01	3	Thacher	NULL	Processed
2019-01-30	1	Elka	NULL	Processed
2018-04-22	10	Elka	Schinner-Predovic	Shipped
2017-07-05	9	Levy	Hettinger LLC	Shipped
2018-09-22	7	Ines	Mraz, Renner and Nolan	Shipped
2017-08-25	5	Clemmie	Satterfield LLC	Shipped
2018-08-02	2	Ilene	Mraz, Renner and Nolan	Shinned

Zadanie 5 - klauzula USING

- Z bazy **sql_invoicing** wyszukaj : (payments).date, client_id, amount, oraz (payment_methods).name.
 - Tam gdzie to możliwe, zastosuj klauzulę USING.

Result Grid | Filter Rows: Search | Export:

	date	client_id	amount	name	
▶	2019-02-12	5	8.18	Credit Card	
	2019-01-03	1	74.55	Credit Card	
	2019-01-11	3	0.03	Credit Card	
	2019-01-26	5	87.44	Credit Card	
	2019-01-15	3	80.31	Credit Card	
	2019-01-15	3	68.10	Credit Card	
	2019-01-08	5	32.77	Credit Card	
	2019-01-08	5	10.00	Cash	

Dzień 2 - Agregowanie Danych

Zadania

Zadanie 1 - funkcje agregujące

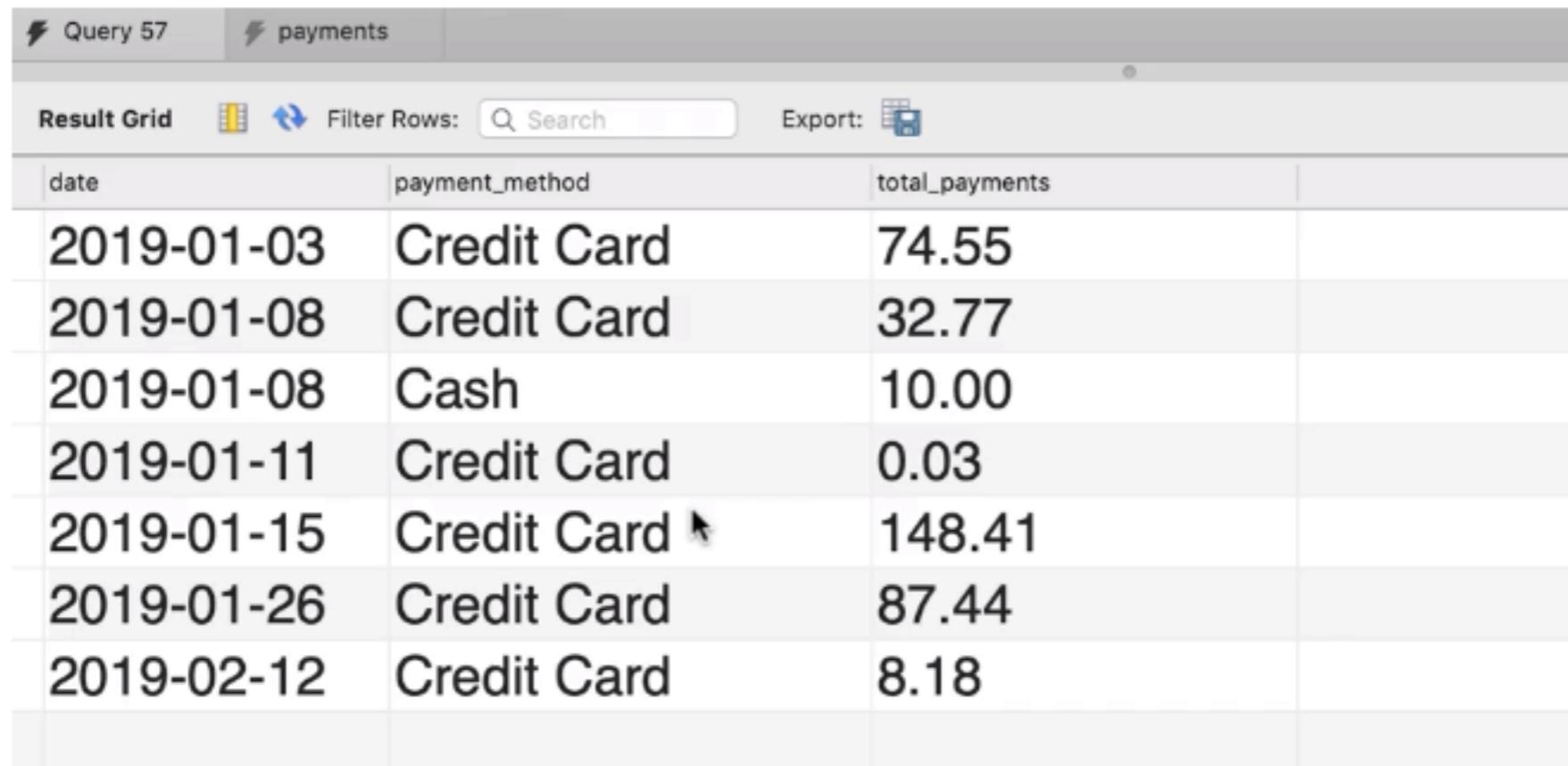
- W bazie **sql_invoicing** stwórz zapytanie, które da rezultat jak poniżej :

The screenshot shows a database query results grid titled "Query 57". The grid has four columns: "date_range", "total_sales", "total_payments", and "what_we_expect". The data is as follows:

date_range	total_sales	total_payments	what_we_expect
First half of 2019	1539.07	662.69	876.38
Second half of 2019	1051.53	355.02	696.51
Total	2590.60	1017.71	1572.89

Zadanie 2 - klauzula GROUP BY

- W bazie **sql_invoicing** stwórz zapytanie, które da rezultat jak poniżej :



The screenshot shows a MySQL Workbench interface with a query results grid. The title bar says "Query 57" and "payments". The grid has three columns: "date", "payment_method", and "total_payments". The data is as follows:

date	payment_method	total_payments
2019-01-03	Credit Card	74.55
2019-01-08	Credit Card	32.77
2019-01-08	Cash	10.00
2019-01-11	Credit Card	0.03
2019-01-15	Credit Card	148.41
2019-01-26	Credit Card	87.44
2019-02-12	Credit Card	8.18

Zadanie 3 - klauzula HAVING

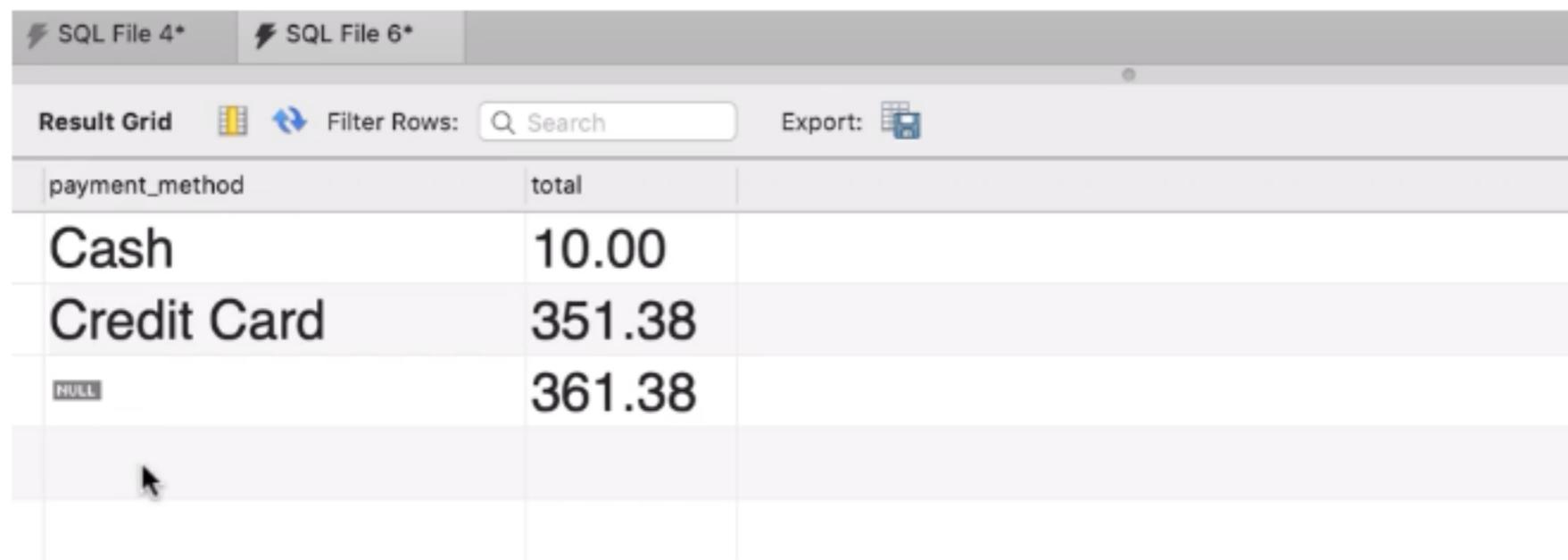
- W bazie **sql_store** stwórz zapytanie, które wyszuka wszystkich klientów ze stanu Wirginia, którzy wydali więcej niż 100\$.
 - Tak aby uzyskać poniższy wynik :

The screenshot shows a MySQL Workbench result grid. The grid has four columns: customer_id, first_name, last_name, and total_sales. There is one row of data: customer_id 2, first_name Ines, last_name Brushfield, and total_sales 157.92. The grid includes standard database navigation buttons (first, previous, next, last) and filtering/searching tools.

customer_id	first_name	last_name	total_sales
2	Ines	Brushfield	157.92

Zadanie 4 - operator WITH ROLLUP

- W bazie **sql_invoicing** na tablicy **payments** stwórz zapytanie, które da rezultat jak poniżej :



The screenshot shows a database query results grid. At the top, there are tabs for "SQL File 4*" and "SQL File 6*". Below the tabs, there are buttons for "Result Grid", "Filter Rows:", "Search" (with a magnifying glass icon), and "Export" (with a document icon). The result grid has two columns: "payment_method" and "total". The data is as follows:

payment_method	total
Cash	10.00
Credit Card	351.38
HULL	361.38

Dzień 2 - Funkcje wbudowane MySQL

Zadania

Zadanie 1 - Funkcje numeryczne

1. Sprawdź działanie funkcji : TRUNCATE, CEILING(), FLOOR(), ABS(), RAND().
2. zobacz resztę funkcji wpisując w google : mysql numeric functions.

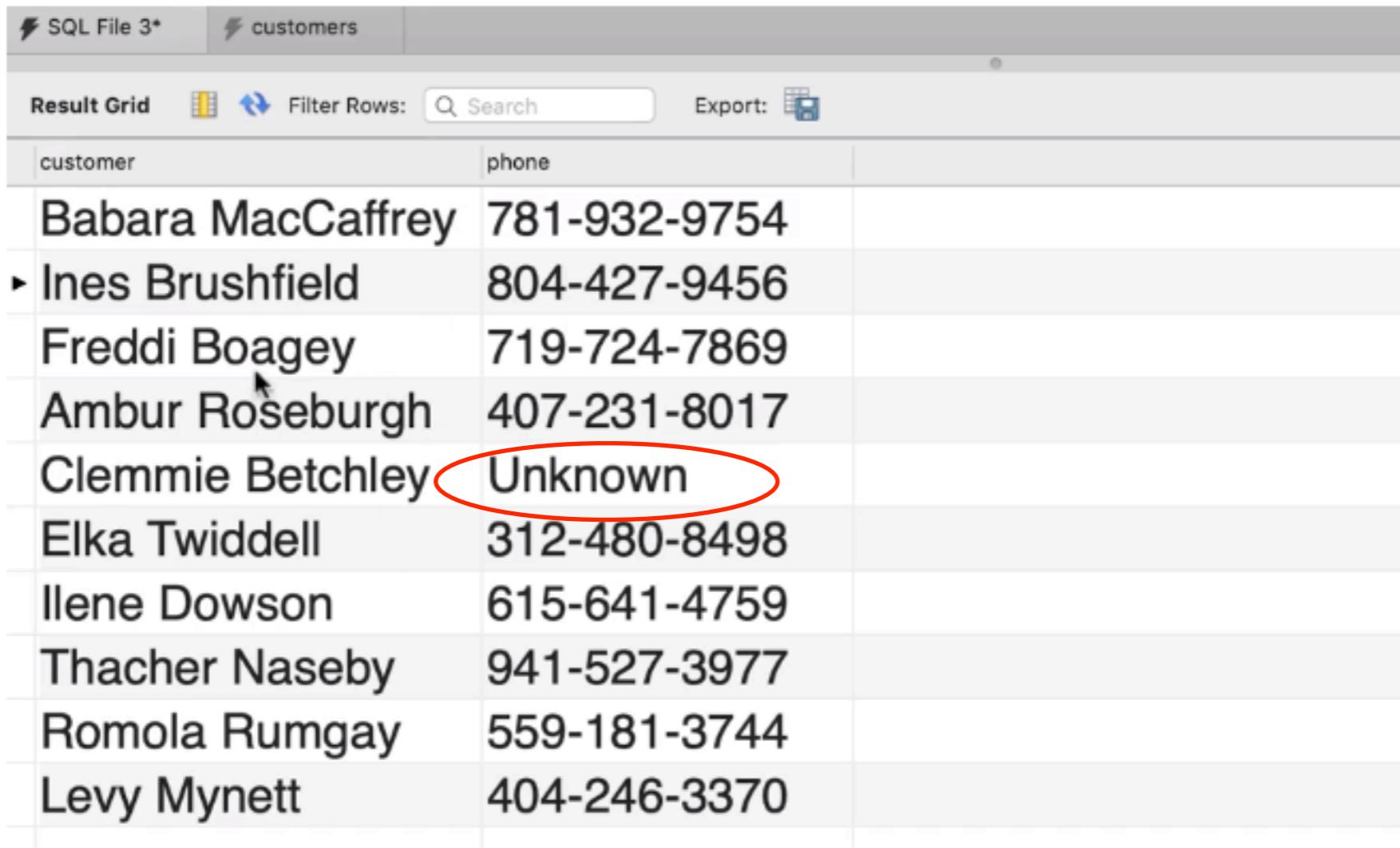
Zadanie 2 -Funkcje daty i czasu

- Popraw poniższe zapytanie, tak aby wskazywało na zamówienia złożone w bieżącym roku (usuń hardcoding).

```
SELECT *  
FROM orders  
WHERE order_date >= '2019-01-01'
```

Zadanie 3 - Funkcje IFNULL() oraz COALESCE()

- W bazie **sql_store** na tablicy **customers** stwórz zapytanie, które w przypadku braku numeru telefonu, wstawi napis “Unknown”.



customer	phone
Babara MacCaffrey	781-932-9754
Ines Brushfield	804-427-9456
Freddi Boagey	719-724-7869
Ambur Roseburgh	407-231-8017
Clemmie Betchley	Unknown
Elka Twiddell	312-480-8498
Ilene Dowson	615-641-4759
Thacher Naseby	941-527-3977
Romola Rumgay	559-181-3744
Levy Mynett	404-246-3370

Zadanie 4 - funkcja IF()

- W bazie **sql_store** stwórz zapytanie, które da rezultat jak poniżej :

product_id	name	orders	frequency
1	Foam Dinner Plate	3	Many times
2	Pork - Bacon,back Peameal	3	Many times
3	Lettuce - Romaine, Heart	4	Many times
4	Brocolinni - Gaylan, Chinese	1	Once
5	Sauce - Ranch Dressing	2	Many times
6	Petit Baguette	2	Many times
8	Island Oasis - Raspberry	1	Once
9	Longan	1	Once
10	Broom - Push	1	Once

Zadanie 5 - operator CASE

- W bazie **sql_invoicing** stwórz zapytanie, które da rezultat jak poniżej :

SQL File 3*

Result Grid Filter Rows: Search Export:

customer	points	category
Clemmie Betchley	3675	Gold
Elka Twiddell	3073	Gold
Freddi Boagey	2967	Silver
Babara MacCaffrey	2273	Silver
Ilene Dowson	1672	Bronze
Romola Rumgay	1486	Bronze
Ines Brushfield	947	Bronze
Levy Mynett	796	Bronze
Ambur Roseburgh	457	Bronze
Thacher Naseby	205	Bronze

Dzień 2 - Widoki (Views)

Materiały dodatkowe i Zadania

Widoki i Tabele

- **Tabela (Table):** jest głównym sposobem do przechowywania danych i informacji w RDBMS. Tabela jest zbiorem powiązanych wpisów danych (rekordów) i składa się z kolumn i wierszy.
- **Widok (View):** - wirtualna tabela, której zawartość jest definiowana przez zapytanie. O ile widok nie jest indeksowany, widok nie istnieje jako przechowywany zestaw wartości danych w bazie danych. Zalety w stosunku do tabeli to:
 1. Możemy łączyć kolumny / wiersze z wielu tabel lub innego widoku i mieć skonsolidowany widok.
 2. Widoki mogą służyć jako mechanizmy bezpieczeństwa, umożliwiając użytkownikom dostęp do danych za pośrednictwem widoku, bez przyznawania użytkownikom uprawnień do bezpośredniego dostępu do bazowych tabel widoku.
 3. Działa jako abstrakcyjna warstwa dla dalszych systemów, więc żadna zmiana w schemacie nie jest ujawniana, a zatem nie ma to wpływu na systemy niższego szczebla.

Zadanie 1 - CREATE VIEW

- W bazie **sql_invoicing** stwórz widok, który dla każdego klienta (clients) będzie pokazywał jego Bilans :
 - client_balance :
 - client_id,
 - name,
 - balance = invoice_total - payment_total.

Dzień 2 - Złożone zapytania

Zadania

Zadanie 1 - Subqueries

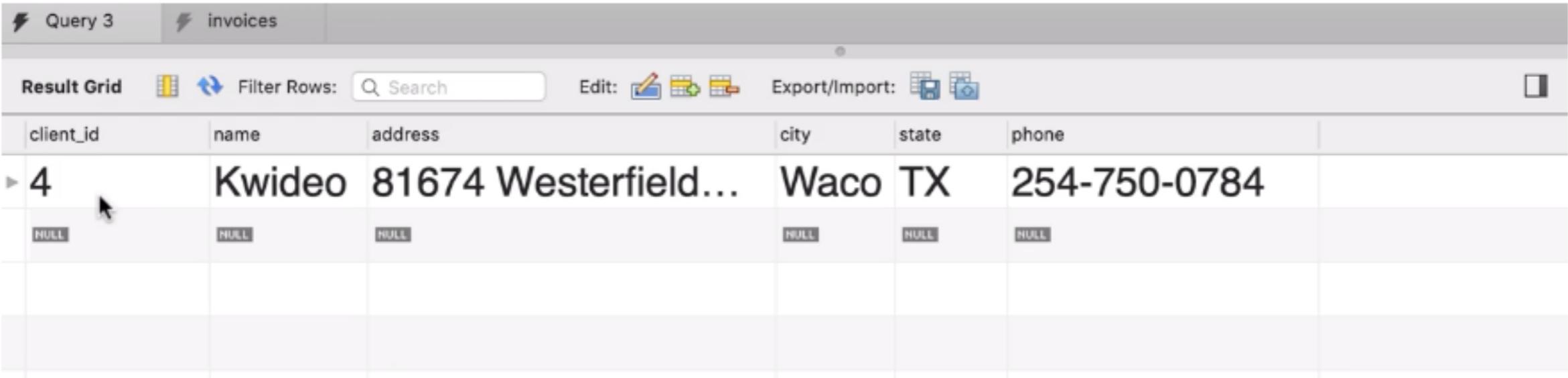
- W bazie **sql_hr**, znajdź pracowników, którzy zarabiają więcej niż średnie wynagrodzenie w firmie.
- Spodziewany wynik :

The screenshot shows a MySQL Workbench interface with a result grid titled "Query 3". The grid displays data from the "employees" table in the "hr" database. The columns shown are employee_id, first_name, last_name, job_title, salary, reports_to, and office_id. The data consists of 11 rows, each representing a different employee with their details. The "job_title" column contains various job descriptions, and the "salary" column shows the corresponding annual salaries.

employee_id	first_name	last_name	job_title	salary	reports_to	office_id
37851	Sayer	Matterson	Statistician III	98926	37270	101
40448	Mindy	Crissil	Staff Scientist	94860	37270	101
56274	Keriann	Alloisi	VP Marketing	110150	37270	101
67009	North	de Clerk	VP Product Manag...	114257	37270	101
67370	Elladine	Rising	Social Worker	96767	37270	101
72540	Guthrey	Iacopetti	Office Assistant I	117690	37270	101
72913	Kass	Hefferan	Computer Systems...	96401	37270	101
76196	Mirilla	Janowski	Cost Accountant	119241	37270	101
84791	Hazel	Tarbert	General Manager	93760	37270	101
95213	Cole	Kesterton	Pharmacist	86119	37270	101
115357	Ivy	Fearey	Structural Engineer	92710	37270	101

Zadanie 2 - klauzula GROUP BY

- W bazie **sql_invoicing** znajdź klientów, którzy nigdy nie mieli wystawionej faktury.
- Spodziewany wynik :

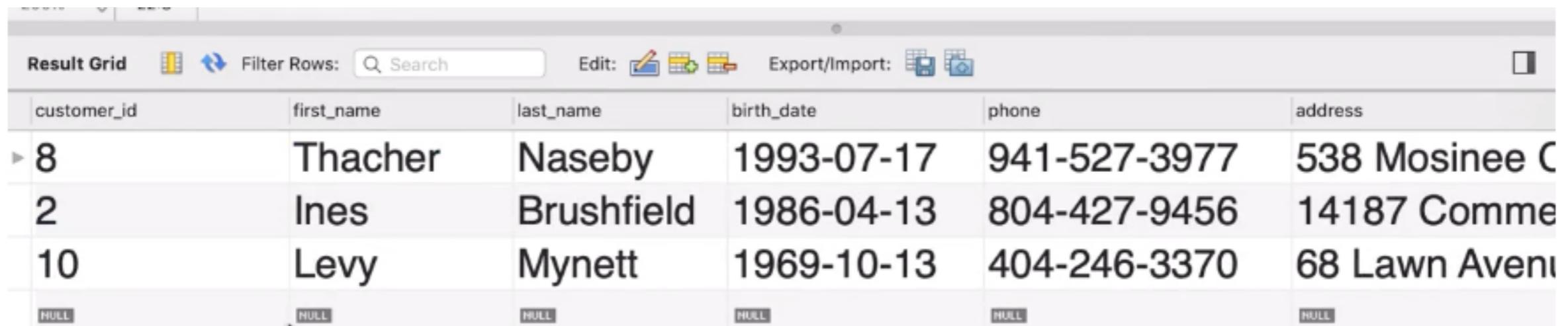


The screenshot shows a MySQL Workbench interface with a query editor titled "Query 3" and a results grid titled "invoices". The results grid displays a single row of data:

client_id	name	address	city	state	phone
4	Kwideo	81674 Westerfield...	Waco	TX	254-750-0784

Zadanie 3 - JOIN czy Subquery ?

- W bazie **sql_store** znajdź klientów, którzy kiedykolwiek zamówili produkt Lettuce (id = 3). Zapytanie skonstruuj z użyciem :
 - a) JOIN oraz
 - b) Subquery.
- Spodziewany wynik :



The screenshot shows a MySQL Workbench interface with a result grid. The grid displays six columns: customer_id, first_name, last_name, birth_date, phone, and address. There are four rows of data. The first row has customer_id 8, first_name Thacher, last_name Naseby, birth_date 1993-07-17, phone 941-527-3977, and address 538 Mosinee C. The second row has customer_id 2, first_name Ines, last_name Brushfield, birth_date 1986-04-13, phone 804-427-9456, and address 14187 Comme. The third row has customer_id 10, first_name Levy, last_name Mynett, birth_date 1969-10-13, phone 404-246-3370, and address 68 Lawn Aveni. The fourth row has all fields set to NULL.

customer_id	first_name	last_name	birth_date	phone	address
8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee C
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Comme
10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Aveni
HULL	HULL	HULL	HULL	HULL	HULL

Zadanie 4 - Operator EXISTS /NOT EXISTS

- W bazie sql_store znajdź produkty, które nigdy nie były zamówione.
 - Zakładamy bardzo duży zbiór produktów.
- Spodziewany wynik :

The screenshot shows a MySQL Workbench result grid. The grid has four columns: product_id, name, quantity_in_stock, and unit_price. The first row contains the data: product_id 7, name 'Sweet Pea Sprouts', quantity_in_stock 98, and unit_price 3.29. All other cells in the grid are empty, indicating null values. The grid has a header row with column names and a toolbar at the top with various database management icons.

product_id	name	quantity_in_stock	unit_price
7	Sweet Pea Sprouts	98	3.29
HULL	HULL	HULL	HULL

Zadanie 5 - Subquery w klauzuli SELECT

- W bazie **sql_invoicing** stwórz zapytanie, które da rezultat jak poniżej :

client_id	name	total_sales	average	difference
1	Vinte	802.89	152.388235	650.501765
2	Myworks	101.79	152.388235	-50.598235
3	Yadel	705.90	152.388235	553.511765
4	Kwideo	NULL	152.388235	NULL
5	Topiclounge	980.02	152.388235	827.631765

Projektowanie baz danych

Materiały pomocnicze i zadania

Modelowanie danych

1. Zrozumienie wymagań biznesowych

- analiza istniejących rozwiązań, formularzy, raportów, interfejsów, wymagań biznesowych, etc...

2. Model konceptualny (koncepcyjny)

- ustalenie zakresu modelu (nie modelujemy Wszechświata !!!)
- uchwycenie głównych obiektów danych i ich relacji - jaką koncepcję chcemy wdrożyć lub pokazać ?
- uzgodnienie modelu z biznesem i interesariuszami

3. Model logiczny

- uzupełnianie modelu konceptualnego, praca z ekspertem domenowym
- dodanie nowych atrybutów/encji, rozwiązanie relacji N : N,
- UWAGA : nie zakładamy, żadnej konkretnej realizacji !! **Praca Analityka / Architekta**

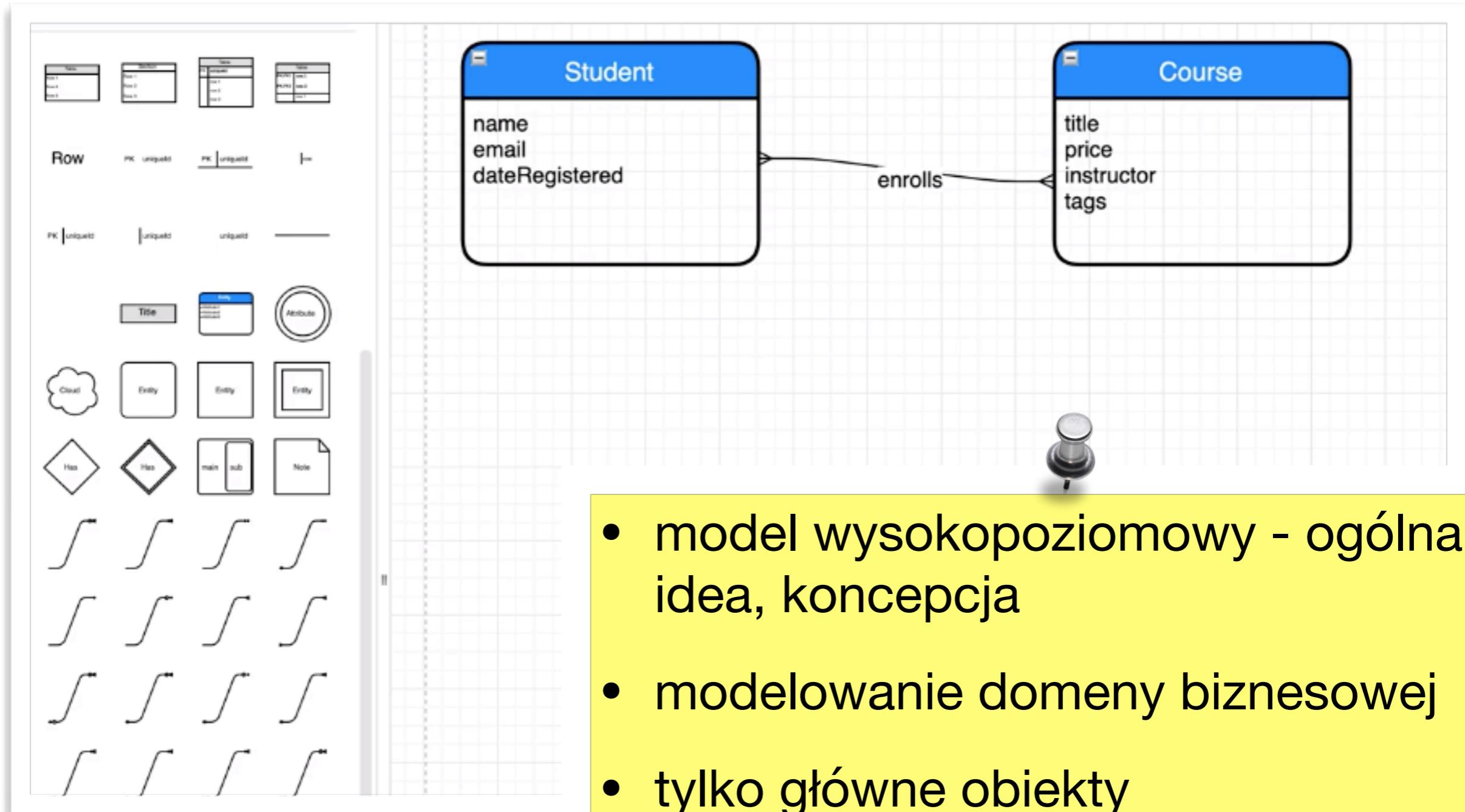
4. Model fizyczny

Praca Developera / Projektanta

- dostosowanie do wymagań danego silnika DB (jak np. MySQL), wyznaczenie kluczy, indeksów, constraints, partycji etc..
- optymalizacja częstych/ krytycznych zapytań, czasem denormalizacja

5. Wygenerowanie skryptu DDL + implementacja na serwerze

Model Konceptualny - zarządzanie kursami

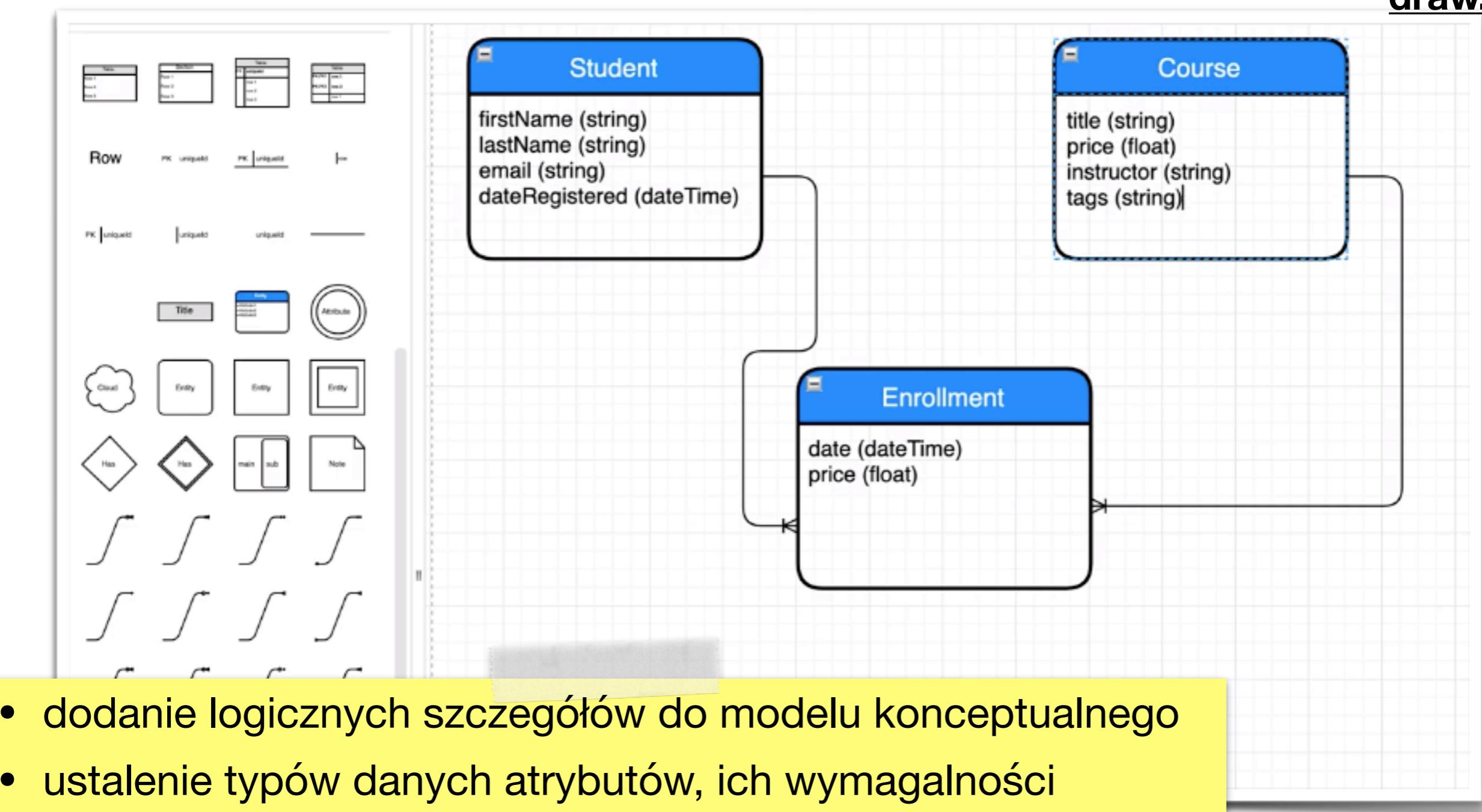


- model wysokopoziomowy - ogólna idea, koncepcja
- modelowanie domeny biznesowej
- tylko główne obiekty
- tylko główne atrybuty

[draw.io](#)

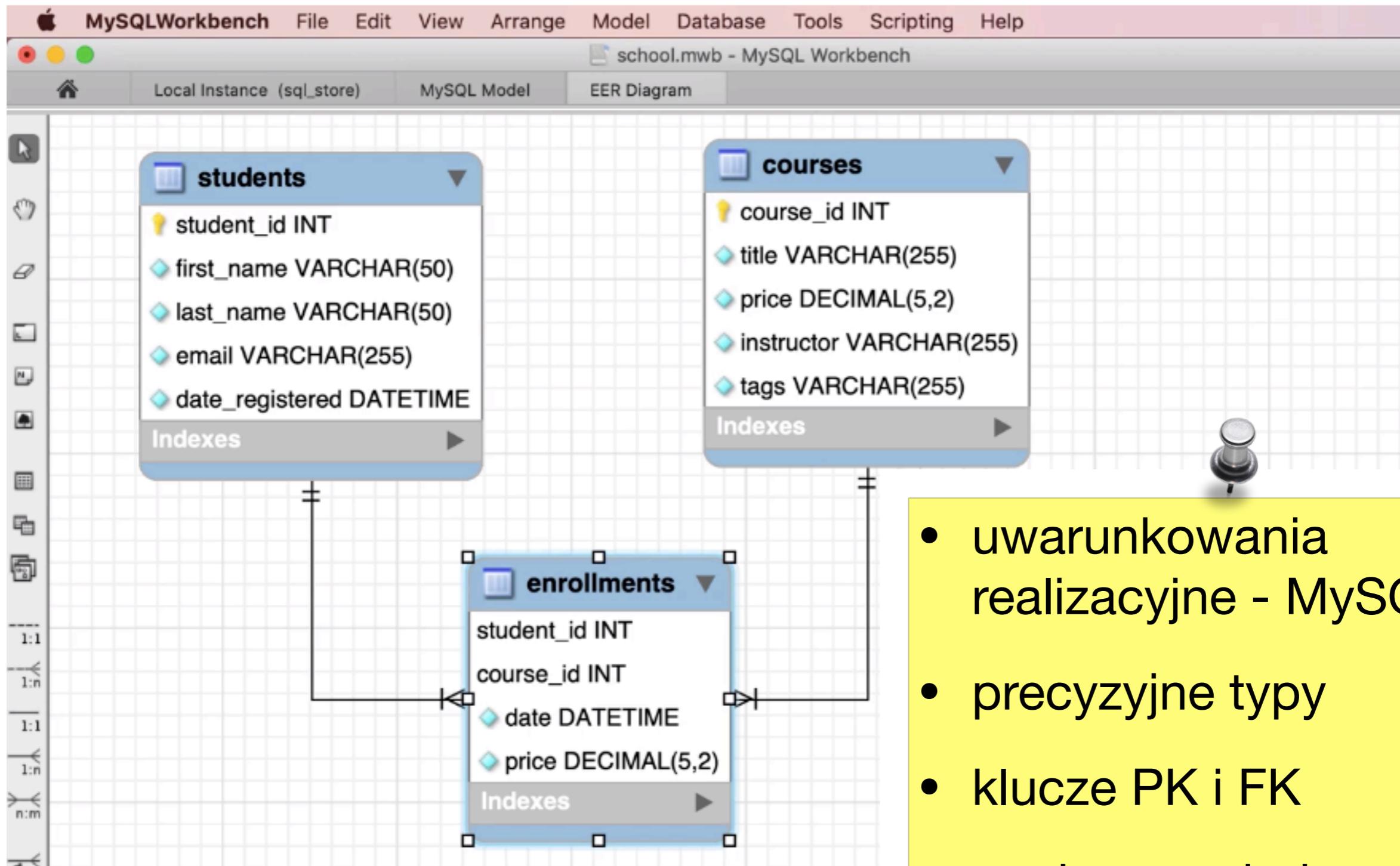
Model Logiczny

[draw.io](#)



- dodanie logicznych szczegółów do modelu konceptualnego
- ustalenie typów danych atrybutów, ich wymagalności
- dekompozycja atrybutów i dodanie dalszych (atryb.)
- rozwiążanie relacji N : N
- dodanie “historii” do obiektu/ relacji

Model Fizyczny



- uwarunkowania realizacyjne - MySQL
- precyzyjne typy
- klucze PK i FK
- atrybuty techniczne
- indeksy

Normalizacja bazy danych

1. Pierwsza postać normalna - 1NF (1st Normal Form)

- Każda komórka tablicy powinna zawierać jedną wartość.
- Każda kolumna w tablicy musi być unikalna (nie może się powtarzać)

2. Druga postać normalna - 2NF (2nd Normal Form)

- Musi spełniać warunki 1NF.
- Każda tablica powinna pojęciowo wyrażać jeden obiekt (byt,entity) oraz każda kolumna w tej tablicy powinna opisywać tylko ten Byt.

3. Trzecia postać normalna - 3NF (3rd Normal Form)

- Musi spełniać warunki 2NF.
- Brak przechodniej zależności funkcjonalnej (ang. transitive functional dependency) *)

*) Przechodnia zależność funkcjonalna występuje, gdy zmiana kolumny niebędącej kluczem może spowodować zmianę dowolnej innej kolumny niebędącej kluczem

Zadanie - system rezerwacji lotów

Los Angeles, CA -> San Francisco, CA
San Francisco, CA -> Los Angeles, CA
Fri Apr 05 2019 -> Sun Apr 07 2019
2 Tickets

Airline Confirmation Numbers

Alaska Airline: TAEGKX

Passengers and Ticket Numbers:

John Smith
Ticket Number: 0177200658

Jennifer Smith
Ticket Number: 0178410326

Price Summary

Traveler 1: \$357.60
Traveler 2: \$357.60
Total: \$756.20

1

Odtwórz model logiczny bazy danych systemu rezerwacji lotów na podstawie poniższych danych z potwierdzenia rezerwacji.

2

Fri Apr 05
Los Angeles -> San Francisco
LAX -> SFO
08:20 AM - 09:35 AM
Alaska Airlines Flight 1490
1h 15m, 236 miles

Depart: Los Angeles Intl Airport (LAX)
Arrive: San Francisco Intl Airport (SFO)

Economy Class

Sun Apr 07
San Francisco -> Los Angeles
SFO -> LAX
02:00 PM - 03:15 PM
Alaska Airlines Flight 1473
1h 15m, 236 miles

Depart: San Francisco Intl Airport (SFO)
Arrive: Los Angeles Intl Airport (LAX)

Economy Class

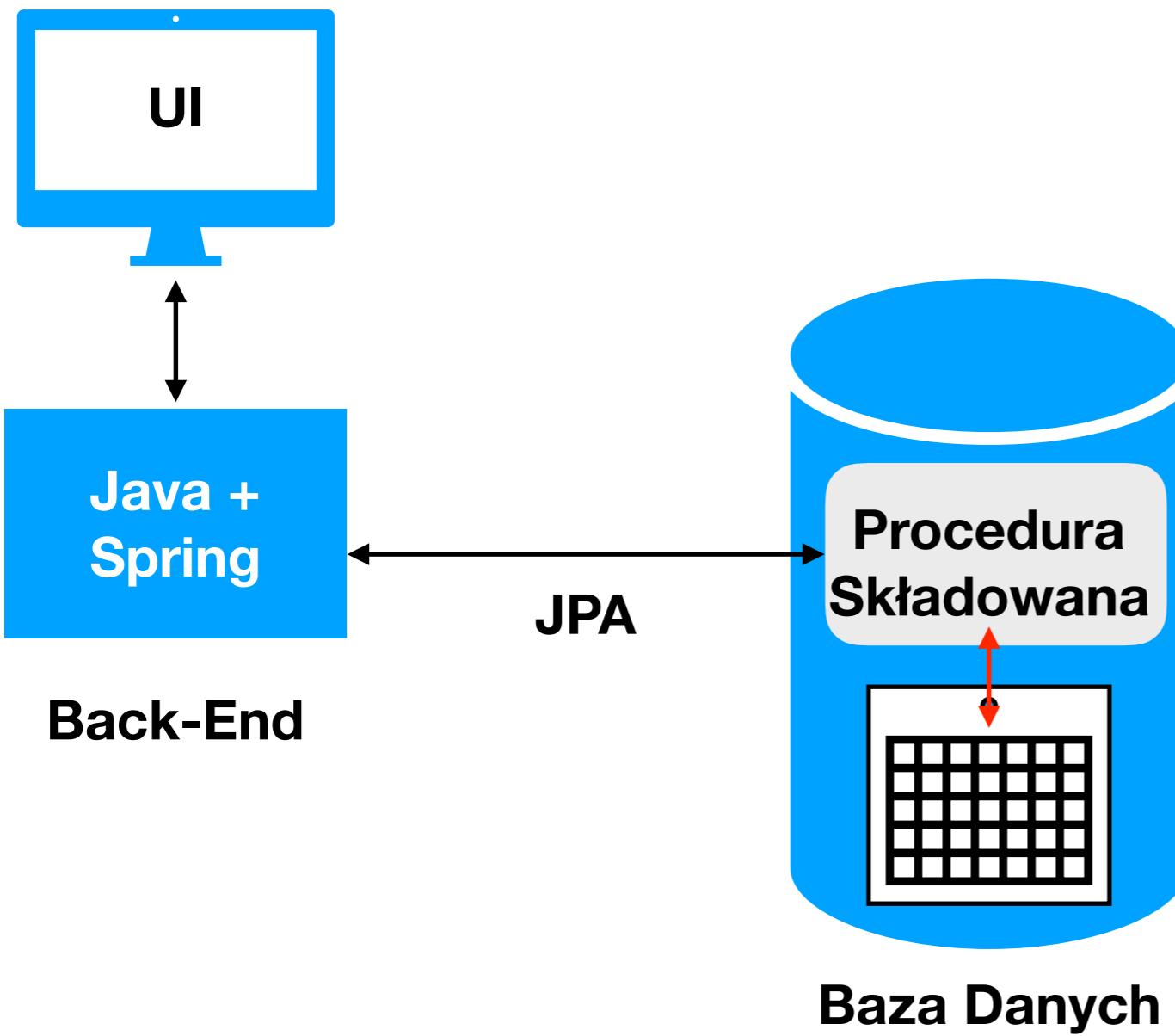
3

Dzień 3 - Procedury składowane

Materiały dodatkowe i Zadania

Procedury składowane - koncepcja

Front-End



1. Umożliwiają przechowywanie i organizowanie kodu po stronie serwera
 - poziom abstrakcji - ukrycie złożoności DB
 - rozdzielenie odpowiedzialności
2. Szybsze wykonanie (lepszy performance)
3. Bezpieczeństwo danych

Zadanie 1 - Tworzymy pierwszą procedurę składowaną

- Utwórz procedurę składowaną `get_invoices_with_balance()`, zwracającą wszystkie faktury dla których `balance > 0`
- Wskazówka :
 - w MySQLWorkbench najpierw stwórz i przetestuj samo zapytanie, a następnie “obuduj” je procedurą
 - pamiętaj o specyfice MySQL, użyj DELIMITER \$\$

Zadanie 2 - Parametry wejściowe procedury

- Napisz w bazie danych sql_invoicing procedurę składowaną zwracającą faktury dla klienta o podanym id.
 - get_invoices_by_client(IN client_id)
 - Użyj generatora kodu MySQL.
- Wskazówka :
 - ustawiamy się w bazie danych na katalogu Stored Procedure
 - prawy przycisk > z menu kontekstowego wybieramy Create Store Procedure.
 - generuje się szkielet procedure w który wpisujemy kod
 - po zakończeniu uruchamiamy przycisk [Apply and Revert]
 - Pojawia się ekran z wygenerowanym kodem (w tym polecenie USE, DELIMITER etc.)
 - Akceptujemy wygenerowany kod - przycisk [Apply]

Dzień 3 - Wyzwalacze i Eventy

Zadania

Zadanie 1 - triggers ON DELETE

- Utwórz kolejny TRIGGER, który wykona operację przeciwną do wykonywanej przez payments_after_insert(), tzn. po usunięciu płatności ZMNIEJSZA saldo (invoices.payment_total) na fakturze.

Dzień 3 - Transakcje i współbieżność

Materiały dodatkowe i zadania

Zadanie - klauzula GROUP BY

- W bazie **sql_invoicing** stwórz zapytanie, które da rezultat jak poniżej :

Dzień 3 - Security

Zadanie 1 - Security

- W bazie **sql_invoicing** :
 1. utwórz nowego użytkownika, mogącego się logować tylko z localhost
 - 2.

