Paweł Szwarnowski, L2, AiBD, grupa czw. 10:15

Zad. 1

Kod: SELECT * FROM category

Odpowiedź: 16

Zad. 2

Kod:

```
SELECT category_id, name FROM category ORDER BY name
```

Odpowiedź:

```
(1, 'Action')

(2, 'Animation')

(3, 'Children')

(4, 'Classics')

(5, 'Comedy')

(6, 'Documentary')

(7, 'Drama')

(8, 'Family')

(9, 'Foreign')

(10, 'Games')

(11, 'Horror')

(12, 'Music')

(13, 'New')

(14, 'Sci-Fi')

(15, 'Sports')

(16, 'Travel')
```

Zad. 3

Kod:

```
SELECT film_id, title, release_year  FROM film ORDER BY release_year ASC LIMIT 1
SELECT film_id, title, release_year FROM film ORDER BY release_year DESC LIMIT 1
```

Odpowiedź:

Każdy z filmów został wydany w 2006 roku, więc nie da się wskazać jednego najnowszego, ani jednego najstarszego filmu. Wynik zwrócony przez polecenia SELECT:

Najstarszy film: (133, 'Chamber Italian', 2006)

Najmłodszy film: (133, 'Chamber Italian', 2006)

Zad. 4

Kod:

```
start_time = datetime.datetime(2005, 7, 1, 0, 0, 0)
end_time = datetime.datetime(2005, 8, 1, 23, 59, 59)
result_set = db.execute(f"SELECT * FROM rental WHERE rental_date >= '{start_time}' AND rental_date <= '{end_time}'")
```

Odpowiedź:

Liczba wypożyczeń między 2005-07-01 a 2005-08-01: 7380

Zad. 5

Kod:

```
start_time = datetime.datetime(2010, 1, 1, 0, 0, 0)
end_time = datetime.datetime(2011, 2, 1, 23, 59, 59)
result_set = db.execute(f"SELECT * FROM rental WHERE rental_date >= '{start_time}' AND rental_date <= '{end_time}'")
```

Odpowiedź:

Liczba wypożyczeń między 2010-01-01 a 2011-02-01: 0

Zad. 6

Kod:

```
result_set = db.execute("SELECT payment_id, amount FROM payment ORDER BY amount DESC LIMIT 1")
```

Odpowiedź:

Największa płatność wypożyczenia: (20403, Decimal('11.99'))

Zad. 7

Kod:

```
result_set = db.execute("SELECT country_id,country FROM country WHERE country IN ('Poland', 'Nigeria', 'Bangladesh')")
for r in result_set:
   cities = db.execute(f"SELECT city_id, city FROM city WHERE country_id = {r.country_id}")
   cities_id = tuple(c.city_id for c in cities)
   addresses = db.execute(f"SELECT address_id FROM address WHERE city_id IN {cities_id}")
   addresses_id = tuple(a.address_id for a in addresses)
   customers = db.execute(f"SELECT * FROM customer WHERE address_id IN {addresses_id}")
```

Odpowiedź:

Customers from Bangladesh:

```
(21, 1, 'Michelle', 'Clark', 'michelle.clark@sakilacustomer.org', 25, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(329, 2, 'Frank', 'Waggoner', 'frank.waggoner@sakilacustomer.org', 334, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(332, 1, 'Stephen', 'Qualls', 'stephen.qualls@sakilacustomer.org', 337, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)
```

Customers from Nigeria:

```
(18, 2, 'Carol', 'Garcia', 'carol.garcia@sakilacustomer.org', 22, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(80, 1, 'Marilyn', 'Ross', 'marilyn.ross@sakilacustomer.org', 84, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(103, 1, 'Gladys', 'Hamilton', 'gladys.hamilton@sakilacustomer.org', 107, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(156, 1, 'Bertha', 'Ferguson', 'bertha.ferguson@sakilacustomer.org', 160, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(198, 2, 'Elsie', 'Kelley', 'elsie.kelley@sakilacustomer.org', 202, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(232, 2, 'Constance', 'Reid', 'constance.reid@sakilacustomer.org', 236, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(250, 2, 'Jo', 'Fowler', 'jo.fowler@sakilacustomer.org', 254, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(277, 2, 'Olga', 'Jimenez', 'olga.jimenez@sakilacustomer.org', 282, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)
```

```
(280, 2, 'Tracey', 'Barrett', 'tracey.barrett@sakilacustomer.org', 285, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(284, 1, 'Sonia', 'Gregory', 'sonia.gregory@sakilacustomer.org', 289, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(286, 1, 'Velma', 'Lucas', 'velma.lucas@sakilacustomer.org', 291, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(409, 2, 'Rodney', 'Moeller', 'rodney.moeller@sakilacustomer.org', 414, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(562, 1, 'Wallace', 'Slone', 'wallace.slone@sakilacustomer.org', 568, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)
```

```
Customers from Poland:

(128, 1, 'Marjorie', 'Tucker', 'marjorie.tucker@sakilacustomer.org', 132, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(270, 1, 'Leah', 'Curtis', 'leah.curtis@sakilacustomer.org', 275, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(318, 1, 'Brian', 'Wyman', 'brian.wyman@sakilacustomer.org', 323, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(380, 1, 'Russell', 'Brinson', 'russell.brinson@sakilacustomer.org', 385, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(501, 1, 'Ruben', 'Geary', 'ruben.geary@sakilacustomer.org', 506, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(558, 1, 'Jimmie', 'Eggleston', 'jimmie.eggleston@sakilacustomer.org', 564, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 0)

(571, 2, 'Johnnie', 'Chisholm', 'johnnie.chisholm@sakilacustomer.org', 577, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)

(572, 1, 'Sidney', 'Burleson', 'sidney.burleson@sakilacustomer.org', 578, True, datetime.date(2006, 2, 14), datetime.datetime(2013, 5, 26, 14, 49, 45, 738000), 1)
```

Zad. 8

Kod:

```
result_set = db.execute("SELECT address_id FROM staff")
addresses_id = tuple(a.address_id for a in result_set)
addresses = db.execute(f"SELECT address, district FROM address WHERE address_id IN {addresses_id}")
```

Odpowiedź:

```
('23 Workhaven Lane', 'Alberta')

('1411 Lillydale Drive', 'QLD')
```

Zad. 9

Kod:

```
result_set = db.execute("SELECT address_id FROM staff")
addresses_id = tuple(a.address_id for a in result_set)
addresses = db.execute(f"SELECT city_id FROM address WHERE address_id IN {addresses_id}")
cities_id = tuple(a.city_id for a in addresses)
cities = db.execute(f"SELECT country_id FROM city WHERE city_id IN {cities_id}")
countries_id = tuple(c.country_id for c in cities)
countries = db.execute(f"SELECT country FROM country WHERE country_id IN {countries_id} AND country IN ('Argentina', 'Spain')")
```

Odpowiedź:

```
Liczba pracowników z Argentyny i Hiszpanii (łącznie): 0
```

Zad. 10

Kod:

```
result_set = db.execute("SELECT return_date, inventory_id FROM rental")
not_returned = tuple(r.inventory_id for r in result_set if r.return_date is None)
film = db.execute(f"SELECT film_id FROM inventory WHERE inventory_id IN {not_returned}")
film_id = tuple(f.film_id for f in film)
category = db.execute(f"SELECT category_id FROM film_category WHERE film_id IN {film_id}")
category_id = tuple(c.category_id for c in category)
names = db.execute(f"SELECT name FROM category WHERE category_id IN {category_id}")
```

Odpowiedź:

```
Wypożyczone kategorie filmów:

Action

Animation

Children

Classics

Comedy

Documentary
```

Drama

Family

Foreign

Games

Horror

Music

New

Sci-Fi

Sports

Travel

Zad. 11

Kod:

```
result_set = db.execute("SELECT country_id FROM country WHERE country IN ('United States', 'Virgin Islands, U.S.', 'Canada', 'Argentina', 'Bolivia', 'Brazil', 'Canada', 'Chile', 'Colombia', 'Dominican Republic', 'Ecuador', 'French Guiana', 'Greenland', 'Mexico', 'Paraguay', 'Peru', 'Puerto Rico', 'Saint Vincent and the Grenadines', 'Venezuela')")
america_id = tuple(r.country_id for r in result_set)
cities = db.execute(f"SELECT city_id, city FROM city WHERE country_id IN {america_id}")
cities_id = tuple(c.city_id for c in cities)
addresses = db.execute(f"SELECT address_id FROM address WHERE city_id IN {cities_id}")
addresses_id = tuple(a.address_id for a in addresses)
stores = db.execute(f"SELECT store_id FROM store WHERE address_id IN {addresses_id}")
store_id = tuple(s.store_id for s in stores)
staff = db.execute(f"SELECT staff_id FROM staff WHERE store_id IN (" + str(store_id[0]) + ")")
staff_id = tuple(s.staff_id for s in staff)
inventory = db.execute(f"SELECT inventory_id FROM rental WHERE staff_id IN (" + str(staff_id[0]) + ")")
inventory_id = tuple(r.inventory_id for r in inventory)
film = db.execute(f"SELECT film_id FROM inventory WHERE inventory_id IN {inventory_id}")
film_id = tuple(f.film_id for f in film)
category = db.execute(f"SELECT category_id FROM film_category WHERE film_id IN {film_id}")
category_id = tuple(c.category_id for c in category)
names = db.execute(f"SELECT name FROM category WHERE category_id IN {category_id}")
```

Odpowiedź:

Action

Animation

Children

Classics

Comedy

Documentary

Drama

Family

Foreign

Games

Horror

Music

New

Sci-Fi

Sports

Travel

Zad. 12

Kod:

```
result_set = db.execute("SELECT actor_id, first_name, last_name FROM actor WHERE
first_name='Olympia' AND last_name='Pfeiffer' OR first_name='Julia' AND last_name='Zellweger' OR
first_name='Ellen' AND last_name='Presley'")
# actor_id = tuple(r.actor_id for r in result_set)
for actor in result_set:
   film = db.execute(f"SELECT film_id FROM film_actor WHERE actor_id={actor.actor_id}")
   film_id = tuple(f.film_id for f in film)
   film_titles = db.execute(f"SELECT title FROM film WHERE film_id IN {film_id}")
```

Odpowiedź:

Tytuły filmów, w których grał/-a Ellen Presley:

Bilko Anonymous

Caribbean Liberty

Casper Dragonfly

Empire Malkovich

Floats Garden

Frogmen Breaking

Homeward Cider

Hyde Doctor

Image Princess

Jacket Frisco

Microcosmos Paradise

Network Peak

Oscar Gold

Pickup Driving

Pinocchio Simon

Private Drop

Roots Remember

Scarface Bang

Secretary Rouge

Spy Mile

Streetcar Intentions

Tadpole Park

Treasure Command

Turn Star

Women Dorado

Tytuły filmów, w których grał/-a Olympia Pfeiffer:

Badman Dawn

Chitty Lock

Color Philadelphia

Contact Anonymous

Deep Crusade

Effect Gladiator

Express Lonely

Firehouse Vietnam

Fugitive Maguire

Hanky October

Idols Snatchers

Ice Crossing

Intolerable Intentions

Mars Roman

Magnolia Forrester

Maude Mod

Murder Antitrust

None Spiking

Others Soup

Psycho Shrunk

Santa Paris

Sense Greek

Storm Happiness

Sweet Brotherhood

Titanic Boondock

Tourist Pelican

Traffic Hobbit

Wait Cider

Tytuły filmów, w których grał/-a Julia Zellweger:

Breakfast Goldfinger

Cranes Reservoir

Dares Pluto

Detective Vision

Divorce Shining

Hollow Jeopardy

Jeopardy Encino

Lambs Cincinatti

Majestic Floats

Minds Truman

Open African

Outlaw Hanky

Panky Submarine

Rider Caddyshack

Won Dares

Wyoming Storm