

Bazy danych

Laboratorium 1

Info

Dane w relacyjnych bazach danych są pogrupowane w relacje (tabele), posiadające atrybuty (kolumny) o wartościach w określonych dziedzinach (typach) i krotki (wiersze) zawierające wartości dla pojedynczego opisywanego obiektu.

Relacja
(tabela)

Atrybut
(kolumna)



NR_AKT	NAZWISKO	STANOWISKO	KIEROWNIK	DATA_ZATR	DATA_ZWOL	PLACA	DOD_FUNKCYJNY	PROWIZJA	ID_DZIALU
1	8901 KROL	PREZES	(null)	89/07/01	(null)	7000	4000	(null)	10
2	8902 MICHALSKI	DYREKTOR	8901	89/08/15	(null)	5000	1500	(null)	40
3	9121 KUKULSKI	DYREKTOR	8901	91/04/02	(null)	5000	1500	(null)	30
4	9011 WIERZBICKI	INFORMATYK	8902	90/03/20	(null)	2000	(null)	(null)	40
5	9235 FIKUS	DYREKTOR	8901	94/09/16	(null)	5000	1500	(null)	70
6	8904 SKALSKI	GLOWNY INFORMATYK	8901	89/08/18	(null)	3500	2500	(null)	10
7	8910 MONIUSZKO	DYREKTOR	8901	89/09/01	(null)	5000	1500	(null)	20
8	8911 WRZOSEK	OPERATOR	8910	89/11/10	(null)	1500	(null)	(null)	20
9	9411 LISIECKI	LABORANT	8910	94/09/10	(null)	1300	(null)	(null)	20
10	8932 BRZOZKA	GLOWNY KSIEGOWY	8901	89/12/06	(null)	3000	2000	(null)	60
11	8913 KOWALSKA	CZLONEK ZARZADU	8901	89/11/15	(null)	5000	2000	(null)	10
12	9010 WISNIEWSKA	GRAFIK	8902	90/02/12	(null)	1800	(null)	(null)	40
13	9025 MALIK	LOGISTYK	9121	90/06/01	(null)	1500	(null)	(null)	30
14	9332 PRUSINSKA	DYREKTOR	8901	97/07/15	(null)	5000	1500	(null)	50

Krotka
(wiersz)

Składnia polecenia SELECT

SELECT [DISTINCT] lista_wyrażeń **FROM** lista_identyfikatorów_tabel
[**WHERE** warunki]
[**GROUP BY** kryteria
[**HAVING** warunki]]
[**ORDER BY** kryteria DESC];

Podstawowe informacje dotyczące SQL

- w poleceniach wielkość liter nie ma znaczenia, w wartościach pól jest rozróżnialna,
- polecenia SQL kończy średnik,
- w celu obejrzenia struktury tabeli należy wydać polecenie:
DESC nazwa_tabeli;
- polecenie zwracające wartości wszystkich atrybutów tabeli:
*SELECT * FROM nazwa_tabeli;*
- wyrażeniami w SQL mogą być:
 - ✓ atrybuty (nazwisko, data_urodzenia, rok, nr_indeksu itp.)
 - ✓ stałe liczbowe (1, 10, 123.321, itp.)
 - ✓ stałe łańcuchowe ('Nowak', 'Kierownik', 'Częstochowa', 'zielony' itp.)
 - ✓ działania arytmetyczne
(placa+dod_funkcyjny+prowizja, placa*12, 5/12, itp.)
 - ✓ konkatencja (połączenie) łańcuchów (imie||' '||nazwisko, itp.)
 - ✓ funkcje operujące na łańcuchach (Substr(imiona, 3, 5), Upper(nazwisko), Lower(kolor), Initcap(nazwisko)), na wartościach liczbowych (Nvl(placa,0)), Round(placa,2), Floor(srednia)) itp.

Rzut (projekcja)

```
SELECT nr_akt, nazwisko, stanowisko, kierownik
FROM pracownicy;
```

Restrykcja (selekcja)

NR_AKT	NAZWISKO	STANOWISKO	KIEROWNIK	DATA_ZATR	DATA_ZWOL	PLACA	DOD_FUNKCYJNY	PROWIZJA	ID_DZIALU
1	8901 KROL	PREZES	(null)	89/07/01	(null)	7000	4000	(null)	10
2	8902 MICHALSKI	DYREKTOR	8901	89/08/15	(null)	5000	1500	(null)	40
3	9121 KUKULSKI	DYREKTOR	8901	91/04/02	(null)	5000	1500	(null)	30
4	9011 WIERZBICKI	INFORMATYK	8902	90/03/20	(null)	2000	(null)	(null)	40
5	9235 FIKUS	DYREKTOR	8901	94/09/16	(null)	5000	1500	(null)	70
6	8904 SKALSKI	GLOWNY INFORMATYK	8901	89/08/18	(null)	3500	2500	(null)	10
7	8910 MONIUSZKO	DYREKTOR	8901	89/09/01	(null)	5000	1500	(null)	20
8	8911 WRZOSEK	STARSZY OPERATOR	8910	89/11/10	(null)	1800	(null)	(null)	20
9	9411 LISIECKI	LABORANT	8910	94/09/10	(null)	1300	(null)	(null)	20
10	8932 BRZOZKA	GLOWNY KSIEGOWY	8901	89/12/06	(null)	3000	2000	(null)	60
11	8913 KOWALSKA	CZLONEK ZARZADU	8901	89/11/15	(null)	5000	2000	(null)	10
12	9010 WISNIEWSKA	GRAFIK	8902	90/02/12	(null)	1800	(null)	(null)	40

```
SELECT * FROM pracownicy
WHERE dod_funkcyjny between 1900 and 2100;
```

Alias

Wewnątrz klauzuli SELECT oraz ORDER BY możliwym jest stosowanie aliasów (przezwoisk) dla wyrażeń:

```
SELECT 'Pracownik:' || Upper(nazwisko) || ' stanowisko:' || Initcap(stanowisko) AS kto, ' zarabia: ' || ((placa+NVL(dod_funkcyjny,0))/1000)||' tys zl.' AS ile FROM pracownicy;
```

KTO	ILE
Pracownik:KROL stanowisko:Prezes	zarabia: 11 tys zl.
Pracownik:MICHALSKI stanowisko:Dyrektor	zarabia: 6,5 tys zl.
Pracownik:KUKULSKI stanowisko:Dyrektor	zarabia: 6,5 tys zl.

Słowo kluczowe **DISTINCT** zapewnia niepowtarzalność otrzymanych w wyniku zapytania wartości.
SELECT DISTINCT imiona FROM studenci;

Wynik zapytania może być uporządkowany według kryterium użytego w klauzuli ORDER BY. Kryterium takim mogą być:

- atrybuty,
- wyrażenia,
- aliasy,
- miejsce na liście wyrażeń klauzuli Select.

Uporządkowanie w odwrotnej kolejności uzyskujemy dodając słowo kluczowe **DESC** po wybranym kryterium.

```
SELECT nr_akt, nazwisko, data_zatr, placa, (placa+NVL(dod_funkcyjny,0)+NVL(prowizja,0)) as pensja from
pracownicy WHERE data_zwol is NULL ORDER BY pensja DESC, 1, data_zatr;
```

NR_AKT	NAZWISKO	DATA_ZATR	PLACA	PENSJA
1	9641 MALYSZ	96/12/05	3000	13500
2	9102 KOWALCZYK	91/01/10	1200	13200
3	8901 KROL	89/07/01	7000	11000
4	9028 LESZCZYNSKI	90/08/10	1200	7200
5	8913 KOWALSKA	89/11/15	5000	7000
6	8902 MICHALSKI	89/08/15	5000	6500
7	8910 MONIUSZKO	89/09/01	5000	6500
8	9121 KUKULSKI	91/04/02	5000	6500
9	9235 FIKUS	94/09/16	5000	6500

Wyróżniamy następujące operatory logiczne:

- porównania: =, !=, <>, <, <=, >, >=, IS NULL, IS NOT NULL, LIKE, NOT LIKE
- logiczne: NOT, AND, OR
- zawarcia w zbiorze: IN
- zawarcia w przedziale: BETWEEN ... AND ...

Istnieje możliwość zastąpienia dowolnego znaku symbolem '_' i ciągu znaków symbolem '%'.
Istnieje możliwość zastąpienia dowolnego znaku symbolem '_' i ciągu znaków symbolem '%'.
Istnieje możliwość zastąpienia dowolnego znaku symbolem '_' i ciągu znaków symbolem '%'.

SELECT imiona, nazwisko, rok, gr_dziekan FROM studenci WHERE rok between 3 and 4 AND gr_dziekan IN(2,4,5) OR (imiona LIKE '_A%' AND nazwisko not LIKE '%K');

Tabele wykorzystywane podczas ćwiczeń (Lab1)

NAZWISKO	NOT NULL	VARCHAR2 (40)
IMIONA	NOT NULL	VARCHAR2 (30)
DATA_URODZENIA		DATE
ADRES		VARCHAR2 (100)
KIERUNEK		VARCHAR2 (50)
SYSTEM		VARCHAR2 (25)
STOPIEN		NUMBER (1)
ROK		NUMBER (1)
SPECJALNOSC		VARCHAR2 (60)
GR_DZIEKAN		NUMBER (2)
SRERDNIA		NUMBER (3, 2)

Rys. 1 DESC STUDENCI;

NR_AKT	NOT NULL	NUMBER (4)
NAZWISKO		VARCHAR2 (20)
STANOWISKO		VARCHAR2 (18)
KIEROWNIK		NUMBER (4)
DATA_ZATR		DATE
DATA_ZWOL		DATE
PLACA		NUMBER (7, 2)
DOD_FUNKCYJNY		NUMBER (7, 2)
PROWIZJA		NUMBER (7, 2)
ID_DZIALU		NUMBER (2)

Rys. 2 DESC PRACOWNICY;

Name	Null	Type
NR_REJESTR	NOT NULL	CHAR (7)
WLASCICIEL		NUMBER (5)
TYP		VARCHAR2 (15)
MARKA		VARCHAR2 (30)
MODEL		VARCHAR2 (30)
KOLOR		VARCHAR2 (30)
POJEMNOSC		NUMBER (4)

Rys. 3 DESC POJAZDY;

Name	Null	Type
ID_KIER	NOT NULL	NUMBER (5)
NAZWISKO		VARCHAR2 (30)
IMIE		VARCHAR2 (30)

Rys. 4 DESC KIEROWCY;

Z cyklu sprawdź jak to działa

```
SELECT * FROM STUDENCI WHERE rok=1 OR rok=2 OR rok=3;
```

```
SELECT * FROM STUDENCI WHERE rok BETWEEN 1 AND 3;
```

```
SELECT * FROM STUDENCI WHERE rok IN (1,2,3);
```

```
SELECT * FROM STUDENCI WHERE rok BETWEEN 1 AND 3 AND (gr_dziekan=1 OR gr_dziekan=3);
```

```
SELECT * FROM STUDENCI WHERE kierunek='INFORMATYKA';
```

```
SELECT * FROM STUDENCI WHERE kierunek LIKE 'INFORMATYKA';
```

```
SELECT nazwisko, imiona, kierunek, rok FROM STUDENCI WHERE kierunek LIKE 'INFORMATYKA' ORDER BY nazwisko, 2 DESC;
```

```
SELECT DISTINCT(kierunek) FROM STUDENCI WHERE kierunek NOT LIKE 'INFORMATYKA';
```

```
SELECT DISTINCT(imiona) FROM STUDENCI WHERE imiona LIKE 'A_a%a';
```

```
SELECT DISTINCT(imiona) FROM STUDENCI WHERE imiona BETWEEN 'K%' AND 'M%' ORDER BY 1;
```

```
SELECT nazwisko||' '||imiona||' studiuje na kierunku '||kierunek Opis FROM STUDENCI;
```

```
SELECT nazwisko||' '||imiona||' studiuje na kierunku '||kierunek as "Podpis kolumny" FROM STUDENCI;
```

```
SELECT nazwisko, dod_funkcyjny, prowizja FROM PRACOWNICY WHERE dod_funkcyjny IS NOT NULL AND prowizja IS NULL;
```

```
SELECT nazwisko, NVL(dod_funkcyjny,0)/placa * 100 as "dodatek/placa w %" FROM PRACOWNICY;
```