

1. Дані і інформація. Їх подання в інформаційних системах.

Данные - это совокупность сведений, зафиксированных на определенном носителе в форме, пригодной для постоянного хранения, передачи и обработки. Преобразование и обработка данных позволяет получить информацию.

Информация - это результат преобразования и анализа данных. Отличие информации от данных состоит в том, что данные - это фиксированные сведения о событиях и явлениях, которые хранятся на определенных носителях, а информация появляется в результате обработки данных при решении конкретных задач. Например, в базах данных хранятся различные данные, а по определенному запросу система управления базой данных выдает требуемую информацию.

Существуют и другие определения информации, например, информация – это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности, неполноты знаний.

Данные представлены посредством строк в таблицах.

- 2. Поняття бази даних, системи баз даних, банка даних.**
- 3. Вимоги до системи баз даних.**
- 4. Структура системи баз даних (СБД), інформаційної системи(ІС).**
- 5. Поняття системи управління базами даних (СКБД). Вимоги до СКБД. Приклади СКБД.**
- 6. Етапи проектування баз даних.**
- 7. Цілі та задачі етапу концептуального проектування бази даних.**
- 8. Поняття об'єкта. Типи об'єктів.**
- 9. Поняття властивості об'єкта. Типи властивостей. Ідентифікація екземпляра об'єкта. Приклади.**
- 10. Поняття зв'язка. Типи зв'язків.**
- 11. Модель "Сутність-Зв'язок".**
- 12. Цілі та задачі етапу логічного проектування бази даних. Основні поняття логічного проектування.**
- 13. Відображення елементів у реляційній моделі даних.**
- 14. Поняття реляційного відношення. Атрибути відношення, кортежи, домени.**
- 15. Поняття ключа відношення. Типи ключів. Контроль цілісності реляційних даних.**

16. Додання, вилучення, поновлення реляційних відношень.

17. Основні недоліки реляційної моделі даних. Приклади.

**18. Поняття функціональної залежності між даними.
Застосування функціональних залежностей.**

19. Часткова і транзитивна функціональні залежності.

20. 1-3 нормальні форми бази даних.

**21. Процедура нормалізації схеми бази даних через
декомпозицію. Приклади.**

22. Означення даних в SQL. Приклади.

23. Оператор SELECT в SQL. Приклади.

рассматривается операция извлечения ин-

формации из СУБД с помощью инструкции SELECT.

Эта информация извлекается из таблиц или представлений
либо для отображения пользователю, либо для внутреннего
использования. Это единственная инструкция в языке за-
просов к данным DQL (Data Query Language).

Использовать инструкцию SELECT относительно про-
сто, однако ее редко выполняют без дополнительных пред-
ложений, и вот именно здесь начинается самое интересное.
Предложения инструкции SELECT можно назвать самыми
запутанными в SQL, и относиться к ним нужно с предель-
ным вниманием.

SELECT [DISTINCT]

{

[<квалификатор>.]<имя_столбца> |

```

* |
<выражение> |
<псевдостолбец>
[AS <псевдоним_столбца>]
},...
FROM
{
<имя_таблицы_или_представления> |
<встроенное_представление>
[[AS] <псевдоним_таблицы>]
}
[WHERE <предикат>]
[GROUP BY [<квалификатор>.]<имя_столбца>,...
[HAVING <предикат>]
]
[ORDER BY {
<имя_столбца> |
<номер_столбца>
}
[ASC | DESC],...
];
SELECT *
FROM status
STATUS_ID_N
-----
6
2
8
9
ST

```

```
--
60
20
70
80
STATUS_DESC_S
-----
SHIPPED
COMPLETE
INVOICED
CANCELLED
```

24. Операторы INSERT, UPDATE, DELETE в SQL. Приклады.

Для добавления в таблицу новых строк либо непосредственно, либо через обновляемое представление используют инструкцию INSERT. Синтаксис этой инструкции несколько отличается в стандарте SQL:2003 и отдельных СУБД (в настоящей книге рассматриваются только Oracle, DB2 9.5 и Microsoft SQL Server 2008). В то же время можно привести такой вид синтаксиса, который будет работать во всех трех ведущих СУБД и не противоречить стандарту SQL:2003.

```
INSERT INTO <имя_таблицы_или_представления>
```

```
[(<имя_столбца>,...)]
```

```
{ {VALUES (<литерал> |
```

```
<выражение> |
```

```
NULL |
```

```
DEFAULT),...} |
```

```
{<инструкция_SELECT>} }
```

```
INSERT INTO product
```

```
(
```

```
prod_id_n,
```

```

prod_num_s,
prod_description_s,
prod_status_s,
prod_brand_s,
prod_pltwid_n,
prod_pltlen_n
)
VALUES
(
991,
'991',
'SPRUCE LUMBER 30X40X50',
'N',
'SPRUCE LUMBER',
4,
6
)

```

Как видите, в инструкции не перечислены имена столбцов PROD_PRICE_N, PROD_NETWEIGHT_N и PROD_SHIPWEIGHT_N, так что их значениями будет NULL.

Для изменения данных таблицы предназначена инструкция UPDATE. Как и в случае с инструкцией INSERT, рассмотренной ранее, обновление в таблице можно выполнить как напрямую, так и посредством обновляемого представления. Ниже приведен общий синтаксис инструкции UPDATE, описанный в стандарте SQL:2003.

```

UPDATE <имя_таблицы_или_представления>
SET <имя_столбца> = { <литерал> |
<выражение> |
(<инструкция_отбора_одной_строки>) |
NULL |
DEFAULT

```

} ,...

[WHERE <предикат>]

UPDATE product

SET prod_pltwid_n = 5,

prod_pltlen_n = 7

WHERE prod_id_n = 990

Инструкция DELETE удаляет строки одной таблицы (напрямую или посредством обновляемого представления). Ее общий синтаксис следующий.

DELETE FROM <имя_таблицы_или_представления>

[WHERE <предикат>]

DELETE FROM salesman

WHERE salesman code s = '02'

25. Поняття транзакції. Властивості транзакції. Реалізація в мові SQL.

26. Механізми підтримки транзакцій в сучасних СКБД. Підтримка розподілених транзакцій.

27. Рівні ізоляції транзакцій. Приклад результатів паралельного виконання транзакцій при різних рівнях ізоляції.

28. Безпека в сучасних СКБД

29. Шифрування даних та його застосування у БД та СКБД

30. Принципи функціонування розподілених БД

31. Розподілені БД, структурні компоненти та їх взаємодія

32. Організація багатокористувацького доступу до БД