

SQL训练题习题答案汇总目录

SQL训练题习题答案汇总目录

- 1.1
- 1.2
- 1.3
- 1.4
- 2.1
- 2.2
- 2.3
- 2.4
- 2.5
- 2.6
- 2.7
- 3.1
- 3.2
- 3.3
- 3.4
- 3.5
- 3.6
- 3.7
- 5.1
- 5.2
- 5.3

SQL训练营页面地址: <https://tianchi.aliyun.com/specials/promotion/aicampsql>
天池龙珠计划训练营地址: <https://tianchi.aliyun.com/specials/promotion/aicamps>

参考答案公布如下:

1.1

```
CREATE TABLE Addressbook
(
    regist_no      INTEGER          NOT NULL,
    name           VARCHAR(128)     NOT NULL,
    address        VARCHAR(256)     NOT NULL,
    tel_no         CHAR(10)         ,
    mail_address   CHAR(20)         ,
    PRIMARY KEY (regist_no));
```

1.2

```
-- [Oracle]
ALTER TABLE Addressbook ADD (postal_code CHAR(8)) NOT NULL;

-- [SQL Server]
ALTER TABLE Addressbook ADD postal_code CHAR(8) NOT NULL;

/*
[DB2] 无法添加。
在DB2中，如果要为添加的列设定NOT NULL约束，
需要像下面这样指定默认值，或者删除NOT NULL约束，
否则就无法添加新列。

*/
-- [DB2 修正版]
ALTER TABLE Addressbook ADD COLUMN postal_code CHAR(8) NOT NULL DEFAULT '0000-000';
```

1.3

```
DROP TABLE Addressbook;
```

1.4

删除后的表无法使用命令进行恢复，请使用习题1.1答案中的 CREATE TABLE 语句再次创建所需的表。

2.1

```
SELECT product_name, regist_date
FROM product
WHERE regist_date > '2009-04-28';
```

2.2

```
/*
下面是问题的SELECT语句
*/
-- ①
SELECT *
FROM product
WHERE purchase_price = NULL;

-- ②
SELECT *
FROM product
```

```
WHERE purchase_price <> NULL;

-- ③
SELECT *
    FROM product
    WHERE product_name > NULL;
```

2.3

```
-- SELECT语句①
SELECT product_name, sale_price, purchase_price
    FROM product
    WHERE sale_price >= purchase_price + 500;

-- SELECT语句②
SELECT product_name, sale_price, purchase_price
    FROM product
    WHERE sale_price - 500 >= purchase_price;
```

2.4

```
SELECT product_name, product_type,
       sale_price * 0.9 - purchase_price AS profit
    FROM product
    WHERE sale_price * 0.9 - purchase_price > 100
       AND ( product_type = '办公用品'
           OR product_type = '厨房用具');
```

2.5

错误 ①

字符型字段 `product_name` 不可以进行 `SUM` 聚合

错误 ②

`WHERE` 语句应该书写在 `GROUP BY` 语句之前（`FROM` 语句之后）

错误 ③

`GROUP BY` 字段（`product_type`）与 `SELECT` 字段不同（`product_id`）

2.6

```
SELECT product_type, SUM(sale_price), SUM(purchase_price)
    FROM product
    GROUP BY product_type
    HAVING SUM(sale_price) > SUM(purchase_price) * 1.5;
```

2.7

```
SELECT *
  FROM product
 ORDER BY regist_date DESC, sale_price;
```

3.1

```
-- 创建视图的语句
CREATE VIEW ViewPractice5_1 AS
SELECT product_name, sale_price, regist_date
  FROM product
 WHERE sale_price >= 1000
       AND regist_date = '2009-09-20';
```

3.2

```
mysql> INSERT INTO ViewPractice5_1 VALUES (' 刀子 ', 300, '2009-11-02');
ERROR 1423 (HY000): Field of view 'shop.viewpractice5_1' underlying table
doesn't have a default value
mysql> desc product;
```

Field	Type	Null	Key	Default	Extra
product_id	char(4)	NO	PRI	NULL	
product_name	varchar(100)	NO		NULL	
product_type	varchar(32)	NO		NULL	
sale_price	int	YES		NULL	
purchase_price	int	YES		NULL	
regist_date	date	YES		NULL	

```
6 rows in set (0.01 sec)
```

解析：插入时将会报错。

视图插入数据时，原表也会插入数据，而原表数据插入时不满足约束条件，所以会报错。（因为 ViewPractice5_1 的原表有三个带有 NOT NULL 约束的字段）

3.3

```
SELECT product_id,
       product_name,
       product_type,
       sale_price,
       (SELECT AVG(sale_price) FROM product) AS sale_price_all
FROM product;
```

3.4

-- 创建视图的语句

```
CREATE VIEW AvgPriceByType AS
SELECT product_id,
       product_name,
       product_type,
       sale_price,
       (SELECT AVG(sale_price)
        FROM product p2
        WHERE p1.product_type = p2.product_type
        GROUP BY p1.product_type) AS avg_sale_price
FROM product p1;
```

-- 确认视图内容

```
SELECT * FROM AvgPriceByType;
```

3.5

正确

3.6

SELECT 查询语句 ①:

```
mysql> SELECT product_name, purchase_price
-> FROM product
-> WHERE purchase_price NOT IN (500, 2800, 5000);

+-----+-----+
| product_name | purchase_price |
+-----+-----+
| 打孔器       | 320           |
| 擦菜板       | 790           |
+-----+-----+
2 rows in set (0.01 sec)
```

解析：该查询语句仅仅取出了 `purchase_price` 不是 500、2800、5000 的商品，而不包含 `purchase_price` 为 `NULL` 的商品，这是因为谓词无法与 `NULL` 进行比较。

SELECT 查询语句 ②:

```
mysql> SELECT product_name, purchase_price
-> FROM product
-> WHERE purchase_price NOT IN (500, 2800, 5000, NULL);
Empty set (0.00 sec)
```

解析：代码执行之前，你可能会认为该语句会返回和查询 ① 同样的结果，实际上它却返回了零条记录，这是因为 `NOT IN` 的参数中不能包含 `NULL`，否则，查询结果通常为空。

3.7

```
SELECT SUM(CASE WHEN sale_price <= 1000 THEN 1 ELSE 0 END) AS
low_price,
      SUM(CASE WHEN sale_price BETWEEN 1001 AND 3000 THEN 1 ELSE 0 END) AS
mid_price,
      SUM(CASE WHEN sale_price >= 3001 THEN 1 ELSE 0 END) AS
high_price
FROM product;
```

5.1

按照 `product_id` 升序排列，计算出截至当前行的最高 `sale_price`。

5.2

如下两种方法都可以实现：

```
-- ①regist_date为NULL时，显示“1年1月1日”。
SELECT regist_date, product_name, sale_price,
      SUM(sale_price) OVER (ORDER BY COALESCE(regist_date, CAST('0001-01-01'
AS DATE))) AS current_sum_price
FROM Product;

-- ②regist_date为NULL时，将该记录放在最前显示。
SELECT regist_date, product_name, sale_price,
      SUM(sale_price) OVER (ORDER BY regist_date NULLS FIRST) AS
current_sum_price
FROM Product;
```

5.3

①:

窗口函数不指定 `PARTITION BY` 就是针对排序列进行全局排序。

②:

本质上是因为 `SQL` 语句的执行顺序。

FROM → WHERE → GROUP BY → HAVING → SELECT → ORDER BY

如果在 `WHERE`, `GROUP BY`, `HAVING` 使用了窗口函数，就是说提前进行了一次排序，排序之后再去除记录、汇总、汇总过滤，第一次排序结果就是错误的，没有实际意义。而 `ORDER BY` 语句执行顺序在 `SELECT` 语句之后，自然是可以使用的。

作者：Datawhale成员：王复振、杨煜、闫钟峰、杨梦迪、苏鹏

如果你对本次学习有任何问题，欢迎加入阿里云天池龙珠计划SQL训练营进行学习交流。

龙珠计划-SQL 训练营

48人



扫一扫群二维码，立刻加入该群。

