SQL应用

第一部分 SQL应用

一、SQL介绍

SQL 结构查询语言(Structured Query Language) 是一个标准化的 广泛使用的语言，可以检索和更新关系表格和数据库中的数据。

二、SQL语法

CREATE语句 生成表;

INSERT、DELETE语句插入和删除行;

UPDATE语句增加或修改在表的列里的数值;

SELECT语句用来检索和操作存于表中的数据;

（子句内的项用逗号分开）

三、语句实例

```
OPTIONS COMPRESS = YES;
```

```
SAS Connection established. Subprocess id is 30779
```

```
34   ods listing close;ods html5 (id=saspy_internal) file=stdout options(bitmap_mode='inline')
device=svg style=HTMLBlue; ods
34 ! graphics on / outputfmt=png;
NOTE: Writing HTML5(SASPY_INTERNAL) Body file: STDOUT
35
36   OPTIONS COMPRESS = YES;
37
38   ods html5 (id=saspy_internal) close;ods listing;

39
```

```
/* CREATE */
PROC SQL;
CREATE TABLE TCUSTR(
    CUSTR_NBR VARCHAR(18),
    SEX INT
);
QUIT;

PROC PRINT DATA = TCUSTR;
RUN;
```

```
54   ods listing close;ods html5 (id=saspy_internal) file=stdout options(bitmap_mode='inline')
device=svg style=HTMLBlue; ods
54 ! graphics on / outputfmt=png;
NOTE: Writing HTML5(SASPY_INTERNAL) Body file: STDOUT
55
56   /* CREATE */
57   PROC SQL;
58   CREATE TABLE TCUSTR(
59       CUSTR_NBR VARCHAR(18),
60       SEX INT
61   );
NOTE: One or more variables were converted because the data type is not supported by the V9 engine.
For more details, run with
      options MSGLEVEL=I.
NOTE: Table WORK.TCUSTR created, with 0 rows and 2 columns.
62   QUIT;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds

63
64   PROC PRINT DATA = TCUSTR;
65   RUN;
NOTE: No observations in data set WORK.TCUSTR.
NOTE: PROCEDURE PRINT used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds

66
67   ods html5 (id=saspy_internal) close;ods listing;

68
```

```
/* INSERT */
PROC SQL;
INSERT INTO
TCUSTR(CUSTR_NBR, SEX)
VALUES("440101200109090011", 1)
VALUES("360101199901010012", 0)
;
QUIT;

PROC PRINT DATA = TCUSTR;
RUN;
```

The SAS System

Obs	CUSTR_NBR	SEX
1	440101200109090011	1
2	360101199901010012	0

```
/* DELETE */
PROC SQL;
DELETE FROM TCUSTR
WHERE CUSTR_NBR = "440101200109090011";
QUIT;

PROC PRINT DATA = TCUSTR;
RUN;
```

The SAS System

Obs	CUSTR_NBR	SEX
2	360101199901010012	0

```
/* UPDATE */
PROC SQL;
UPDATE TCUSTR
SET SEX = 1
WHERE CUSTR_NBR = "360101199901010012";
QUIT;

PROC PRINT DATA = TCUSTR;
RUN;
```

The SAS System

Obs	CUSTR_NBR	SEX
2	360101199901010012	1

四、PROC SQL

SELECT基础结构

SELECT * FROM ACCT;

SQL函数

- COUNT
- SUM
- MAX
- MIN
- AVG
- STD

等等

其他用法

ORDER BY:排序

GROUP BY:分组

WHERE:筛选条件

HAVING:筛选条件(分组后)

```
/* SELECT */
PROC SQL;
SELECT *
FROM SASHELP.CARS(OBS = 5);
QUIT;

PROC SQL;
SELECT *
FROM SASHELP.CARS
WHERE MAKE = "Acura";
QUIT;

PROC SQL;
SELECT MAKE, MSRP
FROM SASHELP.CARS
WHERE MAKE = "Acura";
QUIT;
```

The SAS System

Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	Engine Size (L)	Cylinders	Horsepower	MPG (City)	MPG (Highway)	Weight (LBS)
Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6	265	17	23	4451
Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2	4	200	24	31	2778
Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	4	200	22	29	3230
Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	6	270	20	28	3575
Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	6	225	18	24	3880

The SAS System

Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	Engine Size (L)	Cylinders	Horsepower	MPG (City)	MPG (Highway)	We (L
Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6	265	17	23	4
Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2	4	200	24	31	2
Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	4	200	22	29	3
Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	6	270	20	28	3
Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	6	225	18	24	3
Acura	3.5 RL w/Navigation 4dr	Sedan	Asia	Front	$46,100	$41,100	3.5	6	225	18	24	3
Acura	NSX coupe 2dr manual S	Sports	Asia	Rear	$89,765	$79,978	3.2	6	290	17	24	3

The SAS System

Make	MSRP
Acura	$36,945
Acura	$23,820
Acura	$26,990
Acura	$33,195
Acura	$43,755
Acura	$46,100
Acura	$89,765

```
/* SQL FUNC */
PROC SQL;
SELECT
    COUNT(MSRP),
    SUM(MSRP),
    MAX(MSRP),
    MIN(MSRP),
    AVG(MSRP)
FROM SASHELP.CARS
WHERE MAKE = "Acura";
QUIT;
```

The SAS System

7	300570	89765	23820	42938.57

```
PROC SQL;
SELECT
    COUNT(MSRP) AS CNT_MSRP,
    SUM(MSRP) AS SUM_MSRP,
    MAX(MSRP) AS MAX_MSRP,
    MIN(MSRP) AS MIN_MSRP,
    AVG(MSRP) AS AVG_MSRP
FROM SASHELP.CARS
WHERE MAKE = "Acura";
QUIT;
```

The SAS System

CNT_MSRP	SUM_MSRP	MAX_MSRP	MIN_MSRP	AVG_MSRP
7	300570	89765	23820	42938.57

```
/* GROUP BY */
PROC SQL;
SELECT
    MAKE,
    COUNT(MSRP) AS CNT_MSRP,
    SUM(MSRP) AS SUM_MSRP,
    MAX(MSRP) AS MAX_MSRP,
    MIN(MSRP) AS MIN_MSRP,
    AVG(MSRP) AS AVG_MSRP
FROM SASHELP.CARS(OBS=100)
GROUP BY MAKE;
QUIT;
```

The SAS System

Make	CNT_MSRP	SUM_MSRP	MAX_MSRP	MIN_MSRP	AVG_MSRP
Acura	7	300570	89765	23820	42938.57
Audi	19	822850	84600	25940	43307.89
BMW	20	865705	73195	28495	43285.25
Buick	9	274840	40720	22180	30537.78
Cadillac	8	403795	76200	30835	50474.38
Chevrolet	27	717850	51535	11690	26587.04
Chrysler	10	246235	33295	17985	24623.5

```
/*  ORDER BY*/
PROC SQL;
SELECT
    MAKE,
    COUNT(MSRP) AS CNT_MSRP,
    SUM(MSRP) AS SUM_MSRP,
    MAX(MSRP) AS MAX_MSRP,
    MIN(MSRP) AS MIN_MSRP,
    AVG(MSRP) AS AVG_MSRP
FROM SASHELP.CARS(OBS=100)
GROUP BY MAKE
ORDER BY CNT_MSRP;
QUIT;
```

The SAS System

Make	CNT_MSRP	SUM_MSRP	MAX_MSRP	MIN_MSRP	AVG_MSRP
Acura	7	300570	89765	23820	42938.57
Cadillac	8	403795	76200	30835	50474.38
Buick	9	274840	40720	22180	30537.78
Chrysler	10	246235	33295	17985	24623.5
Audi	19	822850	84600	25940	43307.89
BMW	20	865705	73195	28495	43285.25
Chevrolet	27	717850	51535	11690	26587.04

```
/* WHERE */
PROC SQL;
SELECT
    MAKE,
    COUNT(MSRP) AS CNT_MSRP,
    SUM(MSRP) AS SUM_MSRP,
    MAX(MSRP) AS MAX_MSRP,
    MIN(MSRP) AS MIN_MSRP,
    AVG(MSRP) AS AVG_MSRP
FROM SASHELP.CARS(OBS=100)
WHERE MAKE ^= "Acura"
GROUP BY MAKE
ORDER BY CNT_MSRP;
QUIT;
```

The SAS System

Make	CNT_MSRP	SUM_MSRP	MAX_MSRP	MIN_MSRP	AVG_MSRP
Dodge	2	45905	32235	13670	22952.5
Cadillac	8	403795	76200	30835	50474.38
Buick	9	274840	40720	22180	30537.78
Chrysler	15	408780	38380	17985	27252
Audi	19	822850	84600	25940	43307.89
BMW	20	865705	73195	28495	43285.25
Chevrolet	27	717850	51535	11690	26587.04

```
/* HAVING */
PROC SQL;
SELECT
    MAKE,
    COUNT(MSRP) AS CNT_MSRP,
    SUM(MSRP) AS SUM_MSRP,
    MAX(MSRP) AS MAX_MSRP,
    MIN(MSRP) AS MIN_MSRP,
    AVG(MSRP) AS AVG_MSRP
FROM SASHELP.CARS(OBS=100)
WHERE MAKE ^= "Acura"
GROUP BY MAKE
HAVING CNT_MSRP > 5
ORDER BY CNT_MSRP;
QUIT;
```

The SAS System

Make	CNT_MSRP	SUM_MSRP	MAX_MSRP	MIN_MSRP	AVG_MSRP
Cadillac	8	403795	76200	30835	50474.38
Buick	9	274840	40720	22180	30537.78
Chrysler	15	408780	38380	17985	27252
Audi	19	822850	84600	25940	43307.89
BMW	20	865705	73195	28495	43285.25
Chevrolet	27	717850	51535	11690	26587.04

```
/* CREATE TABLE */
PROC SQL;
CREATE TABLE CARS_GROUP1 AS
SELECT
    MAKE,
    COUNT(MSRP) AS CNT_MSRP,
    SUM(MSRP) AS SUM_MSRP,
    MAX(MSRP) AS MAX_MSRP,
    MIN(MSRP) AS MIN_MSRP,
    AVG(MSRP) AS AVG_MSRP
FROM SASHELP.CARS
WHERE MAKE ^= "Acura"
GROUP BY MAKE
HAVING CNT_MSRP > 5
ORDER BY CNT_MSRP;
QUIT;

PROC PRINT DATA = CARS_GROUP1(OBS=10);
RUN;
```

The SAS System

Obs	Make	CNT_MSRP	SUM_MSRP	MAX_MSRP	MIN_MSRP	AVG_MSRP
1	Porsche	7	584955	192465	43365	83565.00
2	Saab	7	263480	43175	30860	37640.00
3	Saturn	8	137875	23560	10995	17234.38
4	GMC	8	236484	46265	16530	29560.50
5	Suzuki	8	129842	23699	12269	16230.25
6	Infiniti	8	288560	52545	28495	36070.00
7	Cadillac	8	403795	76200	30835	50474.38
8	Buick	9	274840	40720	22180	30537.78
9	Mercury	9	251755	34495	21595	27972.78
10	Lincoln	9	385880	52775	32495	42875.56

```
/* CASE WHEN */
PROC SQL;
CREATE TABLE CARS_GROUP2 AS
SELECT
    MAKE,
    SUM(CASE WHEN ENGINESIZE > 3 THEN MSRP ELSE 0 END) AS SUM_MSRP
FROM SASHELP.CARS
GROUP BY MAKE;
QUIT;

PROC PRINT DATA = CARS_GROUP2(OBS=10);
RUN;
```

The SAS System

Obs	Make	SUM_MSRP
1	Acura	249760
2	Audi	341200
3	BMW	354370
4	Buick	274840
5	Cadillac	403795
6	Chevrolet	589760
7	Chrysler	221615
8	Dodge	229720
9	Ford	355470
10	GMC	219954