Syntax rules for select are as follows :

fdbs-select ::= { fdbs—select-no-group | fdbs-select-group}

fdbs-select-no-group ::= SELECT {\* | list-of-attributes ~~|COUNT(\*)~~ } FROM list-of—tables

[WHERE fdbs-where-clause]

// removed COUNT(\*) because it contradicts page 5: "1. The use pf aggregate functions is limited to statements with a GROUP BY clause."

fdbs-select-group ::= SELECT table.attribute, COUNT(\*) | SUM(attribute) FROM table

GROUP BY table.attribute

list-of— attributes ::=table.attribute [,table.attribute]...

list-of— tables ::= {table | tablel, table2}

fdbs-where-clause ::= {fdbs-where-join-only l fdbs-where-join-and-non-join |

fdbs-where-non—joins}

fdbs-where-join—only ::= (single—join-condition)

single-join-condition ::= tablel .attribute1 comparison tab1e2.attribute2

fdbs-where-join-and-non-joins ::= (single-join-condition) AND (non-join—conditions)

fdbs-Where-non-joins ::= (non-join-condition)

comparison ::= = {= | != | > | >= | < | <=}

non-join—conditions ::= (non-join—condition) [{AND | OR } (non-join-condition)]

non-join—condition ::= tablel.attribute1 comparison constant

should be:

non-join—condition ::= tablel.attribute1 comparison {tablel.attribute2 | constant}

see page 6, paragraph 5. SELECT without Join example:

SELECT \* FROM TABLE1 WHERE (SD = ’Kunz’) OR (SE = 100)

Added )( based on non-join—conditions

constant ::= {integer-constant | string-constant}

**examples**

page 6:

„6. GROUP BY only has to be supported if there is only one table in the FROM clause. In this case there will be exactly be one grouping attribute. The only two aggregate functions you need to support~~ed is~~ are COUNT(\*) and SUM(attribute).

SELECT PERS.ANR, COUNT(\*) FROM PERS GROUP BY PERS.ANR

SELECT PERS.ANR, SUM(SALARY) FROM PERS GROUP BY PERS.ANR

**Concerns on Test Queries :**

Page 4: „To reduce complexity horizontal partitioning will only be required on attributes of data type INTEGER“. But in a CREPARTABS.SQL:

**create** **table** FLUGLINIE (

FLC **varchar**(2),

…

ALLIANZ **varchar**(20) **default** **null**,

**constraint** FLUGLINIE\_LAND\_NN **check** (LAND **is** **not** **null**),

**constraint** FLUGLINIE\_ALLIANZ\_CHK **check** (ALLIANZ **in** 'Star','Excellence','Leftover','OneWorld','SkyTeam')),

**constraint** FLUGLINIE\_FS\_HUB

**foreign** **key** (HUB) **references** FLUGHAFEN(FHC))

HORIZONTAL (FLC('CC','KK'));

horizontal partitioning is applied to VARCHAR.

File: CREPARTABS:SQL

**Concern 1:** You asked not to implement **SET**, in that sense what should our program do if Test Script file provided from you has this script. Same goes for:

**set** echo **on**;

**alter** session **set** nls\_language = english;

**Concern 2:**

**create** **table** FLUGHAFEN (

FHC **varchar**(3),

LAND **varchar**(3),

STADT **varchar**(50) **default** **null**,

NAME **varchar**(50) **default** **null**,

**constraint** FLUGHAFEN\_PS

**primary** **key** (FHC)

);

1. Only capital VARCHAR and INTEGER are to be executed, but Test Script has all keywords including these in small letters.

This has **default** **null** right while defining column. These keywords are neither described in CREATE syntax nor in descriptions.

**Concern 3:**

**create** **table** FLUG (

FNR **integer**,

FLC **varchar**(2),

FLNR **integer**,

VON **varchar**(3),

NACH **varchar**(3),

AB **integer**,

AN **integer**,

**constraint** FLUG\_VON\_NN

**check** (VON **is** **not** **null**),

**constraint** FLUG\_NACH\_NN

**check** (NACH **is** **not** **null**),

**constraint** FLUG\_AB\_NN

**check** (AB **is** **not** **null**),

**constraint** FLUG\_AN\_NN

**check** (AN **is** **not** **null**),

**constraint** FLUG\_AB\_CHK

**check** (AB **between** 0 **and** 2400),

**constraint** FLUG\_AN\_CHK

**check** (AN **between** 0 **and** 2400),

**constraint** FLUG\_VONNACH\_CHK

**check** (VON != NACH),

**constraint** FLUG\_PS

**primary** **key** (FNR),

**constraint** FLUG\_FS\_FLC

**foreign** **key** (FLC) **references** FLUGLINIE(FLC),

**constraint** FLUG\_FS\_VON

**foreign** **key** (VON) **references** FLUGHAFEN(FHC),

**constraint** FLUG\_FS\_NACH

**foreign** **key** (NACH) **references** FLUGHAFEN(FHC)

)

HORIZONTAL (AB(0800,1200));

1. Constraint of check are not tob e implemented, then what to do with different check constraints available.
2. If **foreign** **key** is optional then how parser should react in a test file.

**Concern 4:**

**create** **table** FLUGLINIE (

FLC **varchar**(2),

LAND **varchar**(3),

HUB **varchar**(3) **default** **null**,

NAME **varchar**(30) **default** **null**,

ALLIANZ **varchar**(20) **default** **null**,

**constraint** FLUGLINIE\_LAND\_NN

**check** (LAND **is** **not** **null**),

**constraint** FLUGLINIE\_ALLIANZ\_CHK

**check** (ALLIANZ **in** ('Star','Excellence','Leftover','OneWorld','SkyTeam')),

**constraint** FLUGLINIE\_PS

**primary** **key** (FLC),

**constraint** FLUGLINIE\_FS\_HUB

**foreign** **key** (HUB) **references** FLUGHAFEN(FHC)

)

HORIZONTAL (FLC('CC','KK'));

1. Horizontal has attributes of Strings whereas only Integer are to be parsed as per requirements. Same goes for **check** with **in** and **is** **not** **null**

**CONCERN 5.** How the parser must react to null in **INSERTAIRLINES.SQL:**

**INSERT** **INTO** FLUGLINIE **VALUES** ('AB', 'D ', **null**, 'Air Berlin', **null**);

**update** FLUGLINIE **set** HUB = **NULL**;

**Null** is not a constant defined in grammar as |::= {integer-constant | string-constant}

**CONCERN 6.** COUNT(\*) is an aggregate func used in PARDELS.SQL:

**SELECT** COUNT(\*) **FROM** BUCHUNG;

However, it contradicts page 5: "1. The use pf aggregate functions is limited to statements with a GROUP BY clause."

**CONCERN 7.** This query: **SELECT** FLC, COUNT(\*) **FROM** BUCHUNG **GROUP** **BY** FLC; from PARSELS1OR.SQL has no “BUCHUNG.” table prefix, as required by list-of— attributes ::=table.attribute [,table.attribute]... grammar. What must be the behavior of the parser in such case?

**CONCERN 8.** PARSELS1T.SQL no parentheses after WHERE in **SELECT** \* **FROM** FLUGLINIE **WHERE** FLC = 'BA';

**CONCERN 9.** PARSELS1TGP.SQL has **SELECT** FLC, COUNT(\*) **FROM** FLUG **WHERE** (VON = 'FRA') **GROUP** **BY** FLC; that **WHERE** contradicts

fdbs-select-group ::= SELECT table.attribute, COUNT(\*) | SUM(attribute) FROM table

GROUP BY table.attribute