

ffj SQL

BDE i Borland Database Engine, Å, í Cff SQL i Structured Query Language, ð'É, ¶, Åffj, ", æ, ÑfŠf, [fg, lff[f^fx[fXfe[fuf<, ÉfAfNfZfX, Å, «, Ü, ·Bffj SQL i ufNf%ofCfAf"fgfx[fX SQL v, Æ, àCEÀ, î, ê, Ü, ·j, í ANSI •W€ SQL, lftfufZfbfg, ÅCParadox, ", æ, Ñ dBASE, lfe[fuf<, âftfB[f·fh[iSQL, Å, íu—ñv, ÆCEÀ, î, ê, Ü, ·j, ì-½-¼<K'¥, ðftfj[fg, ·, é, æ, x, ÉŠg'£, ³, ê, Å, ç, Ü, ·B

ffj SQL, Å, í CSQL, ðŽg, Å, ÅufŠf, [fgv, ì SQL fT[fo[[¾, -, Å, È, - Cff[f^fx[fXfT[fo[[ä, É, È, çuf[fj<v, l•W€ff[f^fx[fXfe[fuf<i, ½, Æ, i, í Paradox, Æ dBASE, lfe[fuf<], É, à-â, ç, í, ¹, ðs, x, ±, Æ, ², Å, «, Ü, ·B, Ü, ½ffj SQL, í Cffj, l•W€fe[fuf<, ÆfŠf, [fg, ì SQL fT[fo[[ä, lfe[fuf<, ì-¼•ù, É, Ü, ½, ², éf}f·f`fe[fuf<-â, ç, í, ¹, ðs, x, ½, ß, È, à•K—v, Å, ·B

-½-¼<K'¥

, ±, lfgfsfbfN, Å, í Cff SQL, Å, lfe[fuf<, Æ—ñ, ì-½-¼<K'¥C\•¶, ìŠg'£, Æ\$CEÀ, É, Å, ç, Åà-¾, µ, Ü, ·B

SQL •¶, í Cff[f^è<`CE¾CEêiDDLj, Æff[f^€iCE¾CEêiDMLj, ì, Q, Å, É•ª—p, ³, ê, Ü, ·B

ff[f^€i

fe[fuf<ff[f^,)'¹'ðC}'"üC[XVQC, ", æ, Ñííœ, ÉŽg, í, ê, éff[f^€iCE¾CEêiDMLj, ì SQL •¶B, ±, lfgfsfbfN, Å, í C\•¶, ÆŽg—p—á, É, Å, ç, Åà-¾, µ, Ü, ·B

ff[f^è<`

fe[fuf<, ìi—C•íXCíœC, ", æ, ÑfCf"ffbfNfX, ìi—, Æííœ, ÉŽg, í, ê, éff[f^è<`CE¾CEêiDDLj, ì SQL •¶BDDL, í BDE i Borland Database Enginej, ìŠÖ"CEÀ, Ño, µ, É'¼Ú•İŠ, ³, ê, Ü, ·B, ±, lfgfsfbfN, Å, í C\•¶, ÆŽg—p—á, É, Å, ç, Åà-¾, µ, Ü, ·B

ANSI •W€ SQL, ìÚx, É, Å, ç, Å, í CZs"ì, ì SQL, ì%ðà', È, Ç, ðŽQÆ, µ, Å, , ¾, ³, çB

{button ,AL(` bdedocs')} BDE, ÉŠÖ, ·, éfgfsfbfN

-1/2-1/4<K'¥

ANSI •W€ SQL ,Å,Í□Cfe□[fuf<-1/4,â-ñ-1/4,É,Í
%op□"Žš,ÆfAf"f_□[fXfRfA□i_□j,©,ç,È,é,PCEê,μ,©Žg,!,Ü,¹,ñ□Bf□□[fjf< SQL
,Å,Í□C,æ,è,í,©,è,â,·,ç-1/4'O,ªŽg,!,é,æ,κ,ÉŠg'£,³,ê,Ä,ç,Ü,·□B

fe□[fuf<

f□□[fjf< SQL ,Å,Í□Cfe□[fuf<-1/4,É,Ä,ç,Äftf@fCf<,ÆfpfX,ìŽw'è,ªŠ®'S,Éftf|□[fg,³,è,Ä,ç
,Ü,·□B,1/2,3/4,μfpfX,âftf@fCf<-1/4Šg'£Žq,ðŽ□,Åfe□[fuf<-1/4,Í□C'P^ê^ø-p•,,,Ü,1/2,Í"ñ□d^ø-
p•,,,Å^í,Ü,È,·,é,í,È,è,Ü,¹,ñ□B,1/2,Æ,!,Í□CŽŸ,!,æ,κ,É,μ,Ü,·□B

```
SELECT * FROM 'PARTS.DBF'  
SELECT * FROM "C:\SAMPLE\PARTS.DBF"
```

f□□[fjf< SQL ,Å,Ífe□[fuf<-1/4,É,Ä,ç,Ä BDE ,!fGfŠfAfX,àftf|□[fg,³,è,Ä,ç,Ü,·□B,1/2,Æ,!,Í□CŽŸ,!,æ,κ
,É,μ,Ü,·□B

```
SELECT * FROM ":PDOX:TABLE1"
```

f□□[fjf<fe□[fuf<-1/4,ìŠg'£Žq,ð□È-ª,·,é,Æ□CBDE
SÅ<<□'èft□[fefBfŠfefB,ìfVfXfef€fy□[fW,É, ,é□uDefault Driver□v,ì□Y'è,ÅŽw'è,³,è,Ä,ç
,é,à,í,È,É,è,Ü,·□B, ,é,ç,í-â,ç□‡,í,¹,Ü,1/2,Ífe□[fuf<,ÉCE<,Ñ•t,·,ç,è,Ä,ç,é•W□€fGfŠfAfX,ìffftfHf<fgfhf
%ofCfo,Æ"-,¶Ží-p,!,à,!,É,È,è,Ü,·□B

,Ü,1/2f□□[fjf< SQL ,Å,Í□C'P^ê^ø-p•,,,Ü,1/2,Í"ñ□d^ø-p•,,,Å^í,ß,Í□CSQL fL□[f□□[fh,Æ"-,¶-
1/4'O,ðfe□[fuf<-1/4,Æ,μ,ÄŽg,!,Ü,·□B,1/2,Æ,!,Í□CŽŸ,!,æ,κ,É,μ,Ü,·□B

```
SELECT PASSID FROM "PASSWORD"
```

-ñ

f□□[fjf< SQL ,Å,Í□C^È%º,ì□ðCE□,ð-ž,1/2,μ,Ä,ç,è,Í□CParadox ,!•i□",ìCEê,©,ç,È,é-ñ-1/4□C,·,æ,Ñ
SQL ,!fL□[f□□[fh,Æ"-,¶-1/4'O,ì-ñ-1/4,ªŽg,!,Ü,·□B

- -ñ-1/4,ª'P^ê^ø-p•,,,Ü,1/2,Í"ñ□d^ø-p•,,,Å^í,Ü,è,Ä,ç,é
- -ñ-1/4,ì□æ"ª,É SQL fe□[fuf<-1/4,Ü,1/2,Ífe□[fuf<'ŠŠÖ-1/4,ª•t,·,ç,è,Ä,ç,é
,1/2,Æ,!,Í□CŽŸ,ì-ñ-1/4,í,QCEê,©,ç,È,Ä,Ä,ç,Ü,·□B

```
SELECT E."Emp Id" FROM EMPLOYEE E
```

ŽŸ,ì-á,Å,Í□C-ñ-1/4,í SQL ,! DATE fL□[f□□[fh,Æ"-,¶,Ä,·□B

```
SELECT DATELOG."DATE" FROM DATELOG
```

ŽQ□Æ: -\-ñCEê

—\-ñĈÊ

^È%°º,í□Cf□□[f]f< SQL ,ì—\-ñĈÊ,ìfAf<ftf@fxfbfg□#,ì^ê—,Å,·□B

ACTIVE, ADD, AFTER, ALL, ALTER, AND, ANY, AS, ASC, ASCENDING, AT, AUTO, AUTOINC, AVG

BASE_NAME, BEFORE, BEGIN, BETWEEN, BLOB, BOOLEAN, BOTH, BY, BYTES

CACHE, CAST, CHAR, CHARACTER, CHECK, CHECK_POINT_LENGTH, COLLATE, COLUMN, COMMIT, COMMITTED, COMPUTED, CONDITIONAL, CONSTRAINT, CONTAINING, COUNT, CREATE, CSTRING, CURRENT, CURSOR

DATABASE, DATE, DAY, DEBUG, DEC, DECIMAL, DECLARE, DEFAULT, DELETE, DESC, DESCENDING, DISTINCT, DO, DOMAIN, DOUBLE, DROP

ELSE, END, ENTRY_POINT, ESCAPE, EXCEPTION, EXECUTE, EXISTS, EXIT, EXTERNAL, EXTRACT

FILE, FILTER, FLOAT, FOR, FOREIGN, FROM, FULL, FUNCTION

GDSCODE, GENERATOR, GEN_ID, GRANT, GROUP, GROUP_COMMIT_WAIT_TIME

HAVING, HOUR

IF, IN, INACTIVE, INDEX, INNER, INPUT_TYPE, INSERT, INT, INTEGER, INTO, IS, ISOLATION

JOIN

KEY

LEADING, LEFT, LENGTH, LEVEL, LIKE, LOG_BUFFER_SIZE, LOGFILE, LONG, LOWER

MANUAL, MAX, MAXIMUM_SEGMENT, MERGE, MESSAGE, MIN, MINUTE, MODULE_NAME, MONEY, MONTH

NAMES, NATIONAL, NATURAL, NCHAR, NO, NOT, NULL, NUM_LOG_BUFFERS, NUMERIC

OF, ON, ONLY, OPTION, OR, ORDER, OUTER, OUTPUT_TYPE, OVERFLOW

PAGE_SIZE, PAGE, PAGES, PARAMETER, PASSWORD, PIVOT, PLAN, POSITION, POST_EVENT, PRECISION, PRIMARY, PRIVILEGES, PROCEDURE, PROTECTED

RAW_PARTITIONS, RDB\$DB_KEY, READ, REAL, RECORD_VERSION, REFERENCES, RESERV, RESERVING, RETAIN, RETURNING_VALUES, RETURNS, REVOKE, RIGHT, ROLLBACK

SCHEMA, SECOND, SEGMENT, SELECT, SET, SHADOW, SHARED, SINGULAR, SIZE, SMALLINT, SNAPSHOT, SOME, SORT, SQLCODE, STABILITY, STARTING, STARTS, STATISTICS, SUB_TYPE, SUBSTRING, SUM, SUSPEND

TABLE, THEN, TIME, TIMESTAMP, TIMEZONE_HOUR, TIMEZONE_MINUTE, TO, TRAILING, TRANSACTION, TRIGGER, TRANSFORM, TRIM

UNCOMMITTED, UNION, UNIQUE, UPDATE, UPPER, USER, USING

VALUE, VALUES, VARCHAR, VARIABLE, VARYING, VIEW

WAIT, WHEN, WHERE, WHILE, WITH, WORK, WRITE

YEAR

%°%°žžžq:

||, +, -, *, /, <>, <, >, ,□[f]f“f”□j, =, <=, >=, ~=, !=, ^=, (,)

ff[]f^'€[]ì

,ç,,ç,©[]§-ñ,ª, ,è,Ü,.,ª[]Cf[][]fjfk SQL ,Íff[]f^'€[]ì,ì,½,ß,ìŽŸ,ì•¶,ðFTf[][]fg,µ,Ä,ç,Ü,·[]B

- SELECT[]iŠù'¶,ìff[]f^,ð"Ç,ÝŽæ,é[]j
- INSERT[]ife[]fuf<,É[]V<Kff[]f^,ð'Ç%oÁ,·,é[]j
- UPDATE[]iŠù'¶,ìff[]f^,ð•Í[]X,·,é[]j
- DELETE[]ife[]fuf<,©,çŠù'¶,ìff[]f^,ð[]œ<Ž,·,é[]j

^È%oº,ìfgfsfbfN,Á,Í[]Cf[][]fjfk SQL ,ì DML •¶,Á-~—p,Á,«,éŠÖ[]",É,Â,ç,Ä[]à-¾,µ,Ä,ç,Ü,·[]B

- DML •¶'t,ì^ø[]"uŠ·
- []W[]±SÖ[]"
- •¶Žš-ñSÖ[]"
- "ú•tSÖ[]"
- %o%oZZZq
- -à,ç[]±,í,¹,ì[]X[]V

<ì'ì—á,É,Â,ç,Ä,Í[]CŽŸ,ìfgfsfbfN,ðŽQ[]Æ,µ,Ä,,¾,³,ç[]B

- DML ,ì—á

SELECT

SELECT •¶, í□C, P, Â^Èëä, ìfe□[fuf<, ©, çff□[f^, ð“Ç, ÝŽæ, é□Ü, ÉŽg, ç
, Ü, □B•;□”, ìfe□[fuf<, ©, çff□[f^, ð“Ç, ÝŽæ, é SELECT •¶, í□uCE<□#□v, ÆCEÄ, î, ê, Ü, □Bf□□[fjf< SQL
, íŽÝ, í□Ž®, ì SELECT •¶, ðfTfj□[fg, µ, Ä, ç, Ü, □B

```
SELECT [DISTINCT] column_list
FROM table_reference
[WHERE search_condition]
[ORDER BY order_list]
[GROUP BY group_list]
[HAVING having_condition]
[UNION select_expr]
```

‘□<L, ^a, , é□ê□# , ð□œ, ç, Ä□C□ß, í, ·, x, Ä ANSI •W□€ SQL , Æ“ - I, É^µ, í, ê, Ü, □B[]
“à, ì□ß, ífjvfjvfj” , Ä, □B

column_list , É, ìff□[f^, ì“Ç, ÝŽæ, èCE³, ì—ñ, ðŽw’è, µ, Ü, □B, ½, Æ, í, î□CŽÝ, ì•¶, í, Q, Ä, ì—
ñ, ©, çff□[f^, ð“Ç, ÝŽæ, è, Ü, □B

```
SELECT PART_NO, PART_NAME
FROM PARTS
```

SELECT •¶, ìŽg, ç•û, ì□Ú□×, É, Â, ç, Ä, í□C^È%^o, ìfgfsfbfN, ðŽQ□Æ, µ, Ä, , ³/₄, ³, ç□B

- FROM □ß
- WHERE □ß
- ORDER BY □ß
- GROUP BY □ß
- HAVING □ß
- UNION □ß
- ^ÚŽíCE<□#

FROM

FROM table_reference
, table_reference
, table_reference

```
SELECT PART_NO  
FROM "PARTS.DBF"
```

table_reference , table_reference

```
SELECT * FROM PARTS LEFT OUTER JOIN INVENTORY  
ON PARTS.PART_NO = INVENTORY.PART_NO
```

WHERE

WHERE search_condition
PART_NO > 543

```
SELECT * FROM PARTS  
WHERE PART_NO > 543
```

WHERE PART_NO IN (543, 544, 546, 547)

```
SELECT * FROM PARTS  
WHERE PART_NO IN (543, 544, 546, 547)
```

WHERE search_condition
PART_NO IN (543, 544, 546, 547)

ORDER BY

ORDER BY PART_NAME ASC

```
SELECT * FROM PARTS
ORDER BY PART_NAME ASC
```

ORDER BY PART_NO DESC

```
SELECT * FROM PARTS
ORDER BY PART_NO DESC
```

ORDER BY LAST_NAME || ', ' || FIRST_NAME AS FULL_NAME, PHONE

```
SELECT LAST_NAME || ', ' || FIRST_NAME AS FULL_NAME, PHONE
FROM CUSTOMER
ORDER BY FULL_NAME
```

GROUP BY PART_NAME

GROUP BY ¶

GROUP BY ¶,É,íC¶W¶‡ŠÖ¶”,É,Â,ç,Ä—ñ,ð,Ç,ì,æ,α,ÉfOf<¶[fv
%o»,μ,Ä“Ç,ÝŽæ,é,©,ðŽw’è,μ,Ü,·¶B

UNION

UNION

^ÙŽÍĚ<□‡

f□□[fj]f< SQL ,í^Ù,È,Á,½ff□[f^fx□[fXCE`Ž®,ìfe□[fuf<,ìĚ<□‡,đfTf|□[fg,μ,Ä,ç,Û,·□B,±,ì,æ,κ
,ĚĚ<□‡,đ□u^ÙŽÍĚ<□‡□v,ÆĚÄ,Ñ,Û,·□B

fe□[fuf<-¼,đŽw'è,·,é,É,í□CŽŸ,ì,æ,κ,É,μ,Û,·□B

- f□□[fj]f<fe□[fuf<,ì□ê□‡,í□CfGfŠfAfX,Û,½,ífpfX,đŽw'è,μ,Û,·□B

- fŠf,□[fgfe□[fuf<,ì□ê□‡,í□CfGfŠfAfX,đŽw'è,μ,Û,·□B

ŽŸ,ì•¶,í□CParadox fe□[fuf<,Æ dBASE fe□[fuf<,©,çff□[f^,đ“Ç,ÝŽæ,è,Û,·□B

```
SELECT DISTINCT C.CUST_NO, C.STATE, O.ORDER_NO  
FROM "CUSTOMER.DB" C, "ORDER.DBF" O  
WHERE C.CUST_NO = O.CUST_NO
```

BDE fgfŠfAfX,àfe□[fuf<-¼,Æ'g,Ý□‡,í,¹,ÄŽg—p,Å,«,Û,·□B

INSERT

SQL, VALUES, INSERT INTO EMPLOYEE_PROJECT (EMP_NO, PROJ_ID) VALUES (52, 'DGPII');

—á

INSERT INTO EMPLOYEE_PROJECT (EMP_NO, PROJ_ID) VALUES (52, 'DGPII');

INSERT INTO EMPLOYEE_PROJECT (EMP_NO, PROJ_ID) VALUES (52, 'DGPII');

INSERT INTO PROJECTS

SELECT * FROM NEW_PROJECTS

WHERE NEW_PROJECTS.START_DATE >

'1994-06-06';

UPDATE

ANSI • W€ , ì UPDATE • ¶ , É ' Î , · , é □ S - ñ , â Š g ' £ , Í , , è , Ü , ¹ , ñ □ B

DELETE

ANSI • W € , Ì DELETE • ¶ , É ' Î , · , é □ § - ñ , â Š g ' £ , Í , , è , Û , ¹ , ñ □ B

DML • ¶'†,ì^ø□''uŠ•

DML • ¶'†,Å'ì,ì,©,í,è,É•ï□",Ü,½,í^ø□"f}□[ƒj□[□i?
□j,ðŽg,ì,Ü,□B•ï□",ì□æ"ª,É,Í•K,,fRf□f"□i:□j,ª,È,-,ê,î,È,è,Ü,¹,ñ□B,½,Æ,ì,î□CŽŸ,ì,æ,κ,É,μ,Ü,□B

```
SELECT LAST_NAME, FIRST_NAME  
FROM "CUSTOMER.DB"  
WHERE LAST_NAME > :var1 AND FIRST_NAME < :var2
```


WŁŚÖ”

f SQL , ANSI •W SQL , WŁŚÖ” , f^ , Ç, ÝŽæ, è, ÉŽg, , Ü, •B

- SUM() i—ñ“à, , , x, Ä, i”“l, i#CEv, ð<, ß, é
- AVG() i—ñ“à, , , x, Ä, i”ñ NULL ““l, i•½<i, ð<, ß, é
- MIN() i—ñ“à, iÄ—’l, ð<, ß, é
- MAX() i—ñ“à, iÄ’â’l, ð<, ß, é
- COUNT() iŽw’è, , è, ½ðCE, Æ^è’v, , é—ñ“à, l’l, iCEÂ” , ð” , , é

^È%º, ÉŽ, , è, Ä, ç, é, æ, x, ÉC WCEvŠÖ”“à, Å, iŽ®, iŽg—p, àfTf][fg, , è, Ä, ç, Ü, •B

SUM(Field * 10)
SUM(Field) * 10
SUM(Field1 + Field2)

“ú•tŠÖ”

f SQL, íC“ú•t,âžžŠÔ,ìftfB[f<fh,©,ç’P^è,ì”ìftfB[f<fh,ð•a—£,·,é EXTRACT() ŠÖ”,ðTfj
[fg,μ,Ä,ç,Ü,·BEXTRACT ,ì\•¶,í^É%º,ì,Æ,“,è,Å,·B

```
EXTRACT (extract_field FROM field_name)
```

,½,Æ,ì,ÎCŽŸ,ì•¶,í DATE ftB[f<fh,©,ç”N,ì’I,ð’Šo,μ,Ü,·B

```
SELECT EXTRACT(YEAR FROM HIRE_DATE)  
FROM EMPLOYEE
```

,±,ìŠÖ”,ðŽg,Á,ÄCMONTHCDAYCHOURCMINUTEC,“,æ,Ñ SECOND ,à’Šo,Å,«,Ü,·B

fm[fg:EXTRACT ,Í TIMEZONE_HOUR ,“,æ,Ñ TIMEZONE_MINUTE [B,íTfj[fg,μ,Ä,ç,Ü,¹,ñB

%%ŽŽŽq

f[]fj SQL ,í^È%º,ì%%ŽŽŽq,đTf|[]fg,μ,Ä,ç,Ü,·B

Ží—p

%%ŽŽŽq

ŽZp

+ @-@ * /

"äŠr

< > = <> >= <=

IS NULL IS NOT NULL

~_—

AND OR NOT

•ŕŽš—ñ~ACE<

||

,éfe□[fuf<,ìj□[f\<fnf“fhf<,ì□é□‡,Æ,í^Ù,È,è□CfTf|□[fg,³,ê,Ä,ç,Ü,¹,ñ□B

DML, insert

^È%º,ì—á,íC•W□€ff□[f^fx□[fX,É'í,·,é DML •¶,ðŽ!,μ,Ä,ç,Û,·□B

—á 1: UPDATE

```
update goods
  set city = 'Santa Cruz'
  where goods.city = 'Scotts Valley'
```

—á 2: INSERT

```
insert
  into goods ( part_no, city )
  values ( 'aa0094', 'San Jose' )
```

—á 3: DELETE

```
delete
  from goods
  where part_no = 'aa0093'
```

—á 4: SELECT

ŽŸ,ì—á,íCJOIN ,Æ“ ,¶“,«,ð,·,é,à,ì,Æ,μ,Ä SELECT •¶,ª,Ç,ì,æ,æ,ÉfTf|□[fg,³,ê,Ä,ç,é,©,ðŽ!,μ,Ä,ç,Û,·□B

```
select distinct p.part_no, p.quantity, g.city
  from parts p, goods g
  where p.part_no = g.part_no
  and p.quantity > 20
  order by p.quantity, g.city, p.part_no
```

CE<□‡,ð□s,æ SELECT •¶,É,í WHERE □β,ª,È,¯,ê,î,É,è,Û,¹,ñ□BWHERE □β,É,íC“™%º
¿f'fFfbfN,É,“,ç,Ä□,È,,Æ,à,P,Ä,ìftfB□[f<fh,ªŠÛ,Û,ê,Û,·□B

—á 5: insert

•>'I'ð
•>'I'ð-â,ç□‡,í,¹,ªfTf|□[fg,³,ê,Ä,ç,Û,·□BŽŸ,ì—á,í,»,ì□\•¶,ðŽ!,μ,Ä,ç,Û,·□B

```
select p.part_no
  from parts p
  where p.quantity in
  (select i.quantity
   from inventory i
   where i.part_no = 'aa9393')
```

—á 6: GROUP BY

ŽŸ,ì—á,íCGROUP BY □β,ìŽg,ç•û,ðŽ!,μ,Ä,ç,Û,·□B

```
select part_no, sum(quantity) as PQTY
  from parts
  group by part_no
```

fm□[fg:SELECT □β,Ä□WCEv,ð□s,æ□ê□‡,É,íCGROUP BY
□β,ðŽg,Á,Ä□WCEv'í□Û,Æ,É,éftfB□[f<fh,ðfOf<□[fv%º»,,·,é,æ,æ,É,μ,Û,·□B

—á 7: ORDER BY

ŽŸ,ì—á,íCDESCENDING □β,ðŽg,Á,½ ORDER BY □β,ìŽg,ç•û,ðŽ!,μ,Ä,ç,Û,·□B


```
select distinct customer_no  
  from "c:\data\customer"  
  order by customer_no descending
```

ff [f^'è<

f [f]f SQL

,lfe [fuf<,i [] - C • i X C, ",æ,Ñfhf [fbfv] C,È,ç,Ñ,ÉfCf "ffbfNfX,ì [] -,Æfhf [fbfv],ð [s,α
,½,ß,lff [f^'è<`CE¾CEê [iDDL]j,ðfTf [[fg,μ,Ä,ç,Ü, ·]B

frf... [[,afTf [[fg,³,ê,Ä,ç,Ü, ·]B

f [f]f SQL ,Ä,í DDL • ¶'t,ì • i [] " ,í'í,É'uŠ,Ä,«,Ü,¹,ñ []B

^È%º,ì DDL • ¶,afTf [[fg,³,ê,Ä,ç,Ü, ·]B

- CREATE TABLE
- ALTER TABLE
- DROP TABLE
- CREATE INDEX
- DROP INDEX
- CREATE VIEW

<í'í—á,É,Ä,ç,Ä,í [] CŽŸ,lfghsfbfN,ðŽQ []Æ,μ,Ä,,¾,³,ç []B

- DDL ,ì—á

CREATE TABLE

CREATE TABLE , í^È%º, ì[]\$CEÀ•t, «, Åftf|[][fg, ³, ê, Ä, ç, Ü, ·[]B

- fhf[]fCf“ , ÉŠî, Ä, —ñ'è<` , íftf|[][fg, ³, ê, Ä, ç, Ü, ¹, ñ[]B
 - []\$-ñ[]ðCE[], í[]CParadox , É, Ä, ç, Ä, í PRIMARY KEY , ì, Ý[]Ý'è, Å, «, Ü, ·[]BdBASE , É, Ä, ç, Ä, íftf|[][fg, ³, ê, Ä, ç, Ü, ¹, ñ[]B
- , ½, Æ, Ì, Î[]CŽŸ, Ì•¶, Í LAST_NAME —ñ, Æ FIRST_NAME —ñ, É^êŽŸfL[][\$-ñ[]ðCE[], ðŽ[], Ä Paradox fe[][fuf<, ð[]ì[]-, µ, Ü, ·[]B

```
CREATE TABLE "employee.db"
(
  LAST_NAME CHAR(20),
  FIRST_NAME CHAR(15),
  SALARY NUMERIC(10,2),
  DEPT_NO SMALLINT,
  PRIMARY KEY(LAST_NAME, FIRST_NAME)
)
```

, ±, Ì•¶, ð dBASE fe[][fuf<, É“K—p, ·, é, É, Í[]C^êŽŸfL[][‘è<` , ðŽæ, è[]œ, ©, È, ¯, ê, Í, È, è, Ü, ¹, ñ[]B

```
CREATE TABLE "employee.dbf"
(
  LAST_NAME CHAR(20),
  FIRST_NAME CHAR(15),
  SALARY NUMERIC(10,2),
  DEPT_NO SMALLINT
)
```

Paradox , , æ, Ñ dBASE , ìfe[][fuf<, ì[]ì[]-

f[][]fjff< SQL , ðŽg, Ä, Ä Paradox , , æ, Ñ dBASE , ìfe[][fuf<, ð[]ì[]-, ·, é, É, Í[]Cfe[][fuf<, É- ¼'º, ð•t, ¯, é, Æ, «, Éftf@fCf<Šg'£Žq, ðŽw'è, µ, Ü, ·[]B

- ".DB"[]iParadox fe[][fuf<[]j
- ".DBF"[]idBASE fe[][fuf<[]j

f[][]fjff<fe[][fuf<-¼, É, Ä, ç, ÄŠg'£Žq, ð[]È—ª, ·, é, Æ[]CBDE ŠÅ«[]Ý'èft[][fefBfŠfefB, ÌfVfXfef€fy[][fW, Ì Default Driver []Ý'è, ÄŽw'è, ³, ê, Ä, ç, é, à, Ì, Æ“¯, ¶Ží—p, ìfe[][fuf<, ¢[]ì[]-, ³, ê, Ü, ·[]B

CREATE TABLE , Ä, ìff[][f^CE^f}fbfsf“fo

CREATE TABLE , ÄŽg, í, è, éff[][f^CE^, ÉŠÖ, ·, é SQL , Ì\·¶, Æ[]C, » , ÌSeCE^, ¢ BDE , Ä Paradox , , æ, Ñ dBASE , Ì, Ç, ÌCE^, Éf}fbfsf“fo, ³, ê, é, ©, ðŽŸ, Ì\·, ÄŽ!, µ, Ü, ·[]B

SQL []\·¶	BDE ~ _—[]CE^	Paradox	dBASE
SMALLINT	fldINT16	[]@[]“CE^	[]“[]ICE^ (6,0)
INTEGER	fldINT32	“{·[]@[]“CE^	[]“[]ICE^ (11,0)
DECIMAL(x,y)	fldBCD	BCD CE^	-ç'Í%ºž
NUMERIC(x,y)	fldFLOAT	ŽÀ[]“CE^	[]“[]ICE^ (x,y)
FLOAT(x,y)	fldFLOAT	ŽÀ[]“CE^	·, “ @CE^ (x,y)
CHARACTER(n)	fldZSTRING	•¶ŽšCE^	•¶ŽšCE^
VARCHAR(n)	fldZSTRING	•¶ŽšCE^	•¶ŽšCE^
DATE	fldDATE	“ú•tCE^	“ú•tCE^

BOOLEAN	fldBOOL	~_—□CE^	~_—□CE^
BLOB(n,1)	fldstMEMO	f□f,CE^	f□f,CE^
BLOB(n,2)	fldstBINARY	fofCfifŠCE^	fofCfifŠCE^
BLOB(n,3)	fldstFMTMEMO	□'Ž®•t,«f□f,CE^	-č'í%ž
BLOB(n,4)	fldstOLEOBJ	OLE CE^	OLE CE^
BLOB(n,5)	fldstGRAPHIC	fOf %oftfBbfjNCE^	-č'í%ž
TIME	fldTIME	ŽžŠÔCE^	-č'í%ž
TIMESTAMP	fldTIMESTAMP	"ú•tŽžŠÔCE^	-č'í%ž
MONEY	fldFLOAT, fldstMONEY	<àŠzCE^	•,"®CE^ (20,4)
AUTOINC	fldINT32, fldstAUTOINC	fjfEf"f^CE^	-č'í%ž
BYTES(n)	fldBYTES(n)	fofCfgCE^	-č'í%ž

x = □, "x □iŠù'è'l,ífhf%ofCfoCEÅ—L□j

y = □-□"•" □iŠù'è'l,í 0□j

n = fofCfg'· □iŠù'è'l,í 0□j

1-5 = BLOB fTfuf^fCfv □iŠù'è'l,í 1□j

ALTER TABLE

ALTER TABLE *table* ADD *column_name* *data_type* [, ADD *column_name* *data_type* ...]

ALTER TABLE *table* DROP *column_name* [, DROP *column_name* ...]

ALTER TABLE *table* ADD *column_name* *data_type* [*constraints*]

ALTER TABLE *table* DROP *column_name* [*constraints*]

ALTER TABLE *table* ADD *column_name* *data_type* [*constraints*]

ALTER TABLE *table* DROP *column_name* [*constraints*]

ALTER TABLE *table* ADD *column_name* *data_type* [*constraints*]

ALTER TABLE *table* DROP *column_name* [*constraints*]

ALTER TABLE *table* ADD *column_name* *data_type* [*constraints*]

ALTER TABLE *table* DROP *column_name* [*constraints*], ADD *column_name* *data_type* [*constraints*]

DROP TABLE

DROP TABLE ,Í Paradox ,Ü,½,Í dBASE ,)fe[]fuf<,ð[]í[]œ,μ,Ü,·[]B,½,Æ,¡,î[]CŽŸ,ì•¶,Í Paradox fe[]fuf<,ð[]í[]œ,μ,Ü,·[]B

```
DROP TABLE "employee.db"
```

CREATE INDEX

```
CREATE INDEX index_name ON table_name (column [, column ...])
```

```
CREATE INDEX index_name ON "employee.dbf" (LAST_NAME)
```

```
CREATE INDEX index_name ON "employee.dbf" (LAST_NAME)
```

DROP INDEX

SQL, ANSI DROP INDEX
, É, ½, ß, Ì ANSI DROP INDEX
, É, ½, ß, Ì ANSI DROP INDEX
, É, ½, ß, Ì ANSI DROP INDEX

```
DROP INDEX table_name.index_name | PRIMARY
```

PRIMARY EMPLOYEE.DB
, É, ½, ß, Ì ANSI DROP INDEX
, É, ½, ß, Ì ANSI DROP INDEX

```
DROP INDEX "employee.db".PRIMARY
```

dBASE, Paradox
, É, ½, ß, Ì ANSI DROP INDEX
, É, ½, ß, Ì ANSI DROP INDEX

```
DROP INDEX "employee.db".NAMEX
```


CREATE VIEW

```
CREATE VIEW view_name [(column_name [, column_name]...)]  
AS SELECT ...  
WITH CHECK OPTION
```

The image contains a series of garbled characters and symbols, likely representing a corrupted or heavily distorted version of the SQL 'CREATE VIEW' statement. The visible fragments include 'CREATE VIEW', 'AS SELECT', and 'WITH CHECK OPTION', which are standard SQL syntax elements. The rest of the text is a mix of random characters, including letters, numbers, and special characters, making it largely illegible.

DDL ,ì—á

^È%º,ì—á,Í□C•W□€ff□[f^fx□[fX,É'í,·,é DDL •¶,ìŽg,ç•û,ðŽ!,µ,Ä,ç,Ü,·□B

—á 1a: DDL□iDROP TABLE□j

fe□[fuf<-¼,ÉfsfŠfjh□u.□v,ªŠÜ,Ü,ê,Ä,ç,é□ê□‡,Í-¼'O,ð^ø—p•,,,Ä^Í,Ý,Ü,·□B
drop table "c:\data\customer.db"

—á 1b: DDL□iDROP TABLE□j

fe□[fuf<-¼,ÉfsfŠfjh□u.□v,ªŠÜ,Ü,ê,Ä,ç,È,ç□ê□‡,Í^ø—p•,,,ðŽg,κ•K—v,Í, ,è,Ü,¹,ñ□B
drop table clients

—á 2: DDL□iCREATE INDEX□j

create index part on parts (part_no)

Paradox: Paradox

,ì^èŽŸfCf“ffbfNfX,Í□Cfe□[fuf<□ì□—,·,é,Æ,«,É,ì,Ý□ì□—,Å,«,Ü,·□B“ñŽŸfCf“ffbfNfX,Í□C'â•¶Žš,Æ
□—•¶Žš,ì<æ•É,É,□ì□—,³,ê,Ü,·□B

dBASE: dBASE

,ìfCf“ffbfNfX,Í□C•ÜŽç,³,ê,éfcf“ffbfNfX,Æ,µ,Ä□ì□—,³,ê,Ü,·□BŽw'è,³,ê,éfcf“ffbfNfX-¼,ìf^fo-
¼,Ä,·□B

□ã<L^ÈŠO,ìfCf“ffbfNfX,ì□Ú□×,É,Ä,ç,Ä,Í□CBDE API fìf“f%ofCf“fwf<fv,ì□uDbiAddIndex□v
ðŽQ□Æ,µ,Ä,,¼,³,ç□B

—á 3: DDL□iDROP INDEX□j

fCf“ffbfNfX,ð□í□œ,·,é□\•¶,Í□ufe□[fuf<-¼.fCf“ffbfNfX-¼□v,Ä,·□B
drop index parts.part_no

Paradox: Paradox ,ì□ê□‡,ì,Ý□ufe□[fuf<-¼.primary□v,ì^èŽŸfCf“ffbfNfX,ðŽ!,µ,Ü,·□B

drop index parts.primary

