

## Package Deilvery

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b>CS3528_Project</b>	<b>1</b>
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>7</b>
3.1	Hash::_ht Struct Reference . . . . .	7
3.2	AggInfo Struct Reference . . . . .	7
3.3	AggInfo::AggInfo_col Struct Reference . . . . .	7
3.4	AggInfo::AggInfo_func Struct Reference . . . . .	8
3.5	analysisInfo Struct Reference . . . . .	8
3.6	AuthContext Struct Reference . . . . .	8
3.7	AutoincInfo Struct Reference . . . . .	9
3.8	AuxData Struct Reference . . . . .	9
3.9	Bitvec Struct Reference . . . . .	9
3.10	BtCursor Struct Reference . . . . .	10
3.11	BtLock Struct Reference . . . . .	10
3.12	Btree Struct Reference . . . . .	11
3.13	BtreePayload Struct Reference . . . . .	11
3.14	BtShared Struct Reference . . . . .	11
3.15	BusyHandler Struct Reference . . . . .	12
3.16	CellArray Struct Reference . . . . .	12
3.17	CellInfo Struct Reference . . . . .	13
3.18	Client Class Reference . . . . .	13

3.18.1 Detailed Description . . . . .	15
3.18.2 Constructor & Destructor Documentation . . . . .	15
3.18.2.1 Client(const string &nam, const string &add, const string &cit, const string &st, const string &z, const int &clientID) . . . . .	15
3.18.3 Member Function Documentation . . . . .	15
3.18.3.1 getAddress() const . . . . .	15
3.18.3.2 getCoords() . . . . .	15
3.18.3.3 getReceivedPackages() const . . . . .	15
3.18.3.4 getSentPackages() const . . . . .	16
3.18.3.5 parseAddress() . . . . .	16
3.18.3.6 receivePackage(Package *pack) . . . . .	16
3.18.3.7 sendPackage(Package *pack) . . . . .	16
3.18.3.8 setAddress(const string &add, const string &cit, const string &st, const string &z) . . . . .	16
3.18.3.9 toString() const . . . . .	17
3.19 CollSeq Struct Reference . . . . .	17
3.20 Column Struct Reference . . . . .	17
3.21 compareInfo Struct Reference . . . . .	18
3.22 CountCtx Struct Reference . . . . .	18
3.23 With::Cte Struct Reference . . . . .	18
3.24 DateTime Struct Reference . . . . .	19
3.25 Db Struct Reference . . . . .	19
3.26 DbFixer Struct Reference . . . . .	19
3.27 DistinctCtx Struct Reference . . . . .	20
3.28 et_info Struct Reference . . . . .	20
3.29 Expr Struct Reference . . . . .	20
3.30 ExprList Struct Reference . . . . .	21
3.31 ExprList::ExprList_item Struct Reference . . . . .	22
3.32 ExprSpan Struct Reference . . . . .	22
3.33 FileChunk Struct Reference . . . . .	22
3.34 FilePoint Struct Reference . . . . .	23
3.35 FKey Struct Reference . . . . .	23

3.36	<a href="#">fts5_api Struct Reference</a>	24
3.37	<a href="#">fts5_tokenizer Struct Reference</a>	24
3.38	<a href="#">Fts5ExtensionApi Struct Reference</a>	24
3.39	<a href="#">Fts5Phraselter Struct Reference</a>	25
3.40	<a href="#">FuncDef Struct Reference</a>	25
3.41	<a href="#">FuncDefHash Struct Reference</a>	26
3.42	<a href="#">FuncDestructor Struct Reference</a>	26
3.43	<a href="#">Genetic Class Reference</a>	26
3.44	<a href="#">Hash Struct Reference</a>	27
3.45	<a href="#">HashElem Struct Reference</a>	27
3.46	<a href="#">IdList Struct Reference</a>	27
3.47	<a href="#">IdList::IdList_item Struct Reference</a>	28
3.48	<a href="#">IdxCover Struct Reference</a>	28
3.49	<a href="#">ImportCtx Struct Reference</a>	28
3.50	<a href="#">Incrblob Struct Reference</a>	29
3.51	<a href="#">IncrMerger Struct Reference</a>	29
3.52	<a href="#">Index Struct Reference</a>	29
3.53	<a href="#">IndexSample Struct Reference</a>	30
3.54	<a href="#">InitData Struct Reference</a>	30
3.55	<a href="#">IntegrityCk Struct Reference</a>	31
3.56	<a href="#">KeyInfo Struct Reference</a>	31
3.57	<a href="#">LimitVal Struct Reference</a>	32
3.58	<a href="#">Lookaside Struct Reference</a>	32
3.59	<a href="#">LookasideSlot Struct Reference</a>	32
3.60	<a href="#">Mem Struct Reference</a>	33
3.61	<a href="#">MemJournal Struct Reference</a>	33
3.62	<a href="#">MemPage Struct Reference</a>	34
3.63	<a href="#">Mem::MemValue Union Reference</a>	34
3.64	<a href="#">MergeEngine Struct Reference</a>	35
3.65	<a href="#">Module Struct Reference</a>	35

3.66 NameContext Struct Reference . . . . .	35
3.67 VdbeOp::p4union Union Reference . . . . .	36
3.68 Package Class Reference . . . . .	36
3.68.1 Detailed Description . . . . .	37
3.68.2 Constructor & Destructor Documentation . . . . .	37
3.68.2.1 Package(Client *s, Client *r, const float w, const Priority p) . . . . .	37
3.68.3 Member Function Documentation . . . . .	38
3.68.3.1 getPriority() const . . . . .	38
3.68.3.2 getPriorityString() const . . . . .	38
3.68.3.3 getReceiver() const . . . . .	38
3.68.3.4 getSender() const . . . . .	38
3.68.3.5 getWeight() const . . . . .	39
3.68.3.6 setPriority(const Priority) . . . . .	39
3.68.3.7 setReceiver(Client *) . . . . .	39
3.68.3.8 setSender(Client *) . . . . .	39
3.68.3.9 setWeight(const double) . . . . .	39
3.69 Pager Struct Reference . . . . .	40
3.70 PagerSavepoint Struct Reference . . . . .	41
3.71 Parse Struct Reference . . . . .	42
3.72 PCache Struct Reference . . . . .	43
3.73 PCache1 Struct Reference . . . . .	44
3.74 PgFreeslot Struct Reference . . . . .	44
3.75 PgHdr Struct Reference . . . . .	44
3.76 PgHdr1 Struct Reference . . . . .	45
3.77 PGroup Struct Reference . . . . .	45
3.78 PmaReader Struct Reference . . . . .	46
3.79 PmaWriter Struct Reference . . . . .	46
3.80 PreUpdate Struct Reference . . . . .	47
3.81 PrintfArguments Struct Reference . . . . .	47
3.82 randomPackageEnum Struct Reference . . . . .	47

3.83 ReusableSpace Struct Reference . . . . .	48
3.84 RowSet Struct Reference . . . . .	48
3.85 RowSetChunk Struct Reference . . . . .	48
3.86 RowSetEntry Struct Reference . . . . .	49
3.87 SavedModelInfo Struct Reference . . . . .	49
3.88 Savepoint Struct Reference . . . . .	49
3.89 ScanStatus Struct Reference . . . . .	50
3.90 Schema Struct Reference . . . . .	50
3.91 FKey::sColMap Struct Reference . . . . .	50
3.92 ScratchFreeslot Struct Reference . . . . .	51
3.93 Select Struct Reference . . . . .	51
3.94 SelectDest Struct Reference . . . . .	51
3.95 ShellState Struct Reference . . . . .	52
3.96 SortCtx Struct Reference . . . . .	53
3.97 SorterFile Struct Reference . . . . .	53
3.98 SorterList Struct Reference . . . . .	54
3.99 SorterRecord Struct Reference . . . . .	54
3.100SortSubtask Struct Reference . . . . .	54
3.101sqlite3 Struct Reference . . . . .	55
3.102sqlite3_api_routines Struct Reference . . . . .	56
3.103sqlite3_backup Struct Reference . . . . .	61
3.104sqlite3_context Struct Reference . . . . .	61
3.105sqlite3_file Struct Reference . . . . .	62
3.106sqlite3_index_info::sqlite3_index_constraint Struct Reference . . . . .	62
3.107sqlite3_index_info::sqlite3_index_constraint_usage Struct Reference . . . . .	62
3.108sqlite3_index_info Struct Reference . . . . .	62
3.109sqlite3_index_info::sqlite3_index_orderby Struct Reference . . . . .	63
3.110sqlite3_io_methods Struct Reference . . . . .	64
3.111sqlite3_mem_methods Struct Reference . . . . .	64
3.112sqlite3_module Struct Reference . . . . .	65

3.113sqlite3_mutex Struct Reference . . . . .	65
3.114sqlite3_mutex_methods Struct Reference . . . . .	65
3.115sqlite3_pcache_methods Struct Reference . . . . .	66
3.116sqlite3_pcache_methods2 Struct Reference . . . . .	66
3.117sqlite3_pcache_page Struct Reference . . . . .	67
3.118sqlite3_rtree_geometry Struct Reference . . . . .	67
3.119sqlite3_rtree_query_info Struct Reference . . . . .	68
3.120sqlite3_vfs Struct Reference . . . . .	68
3.121sqlite3_vtab Struct Reference . . . . .	69
3.122sqlite3_vtab_cursor Struct Reference . . . . .	69
3.123Sqlite3Config Struct Reference . . . . .	69
3.124sqlite3::sqlite3InitInfo Struct Reference . . . . .	71
3.125SQLiteThread Struct Reference . . . . .	71
3.126SrcCount Struct Reference . . . . .	71
3.127SrcList Struct Reference . . . . .	72
3.128SrcList::SrcList_item Struct Reference . . . . .	72
3.129Stat4Accum Struct Reference . . . . .	73
3.130Stat4Sample Struct Reference . . . . .	73
3.131StrAccum Struct Reference . . . . .	74
3.132SubProgram Struct Reference . . . . .	74
3.133SumCtx Struct Reference . . . . .	74
3.134Table Struct Reference . . . . .	75
3.135TableLock Struct Reference . . . . .	75
3.136TabResult Struct Reference . . . . .	76
3.137Token Struct Reference . . . . .	76
3.138TrigEvent Struct Reference . . . . .	76
3.139Trigger Struct Reference . . . . .	76
3.140TriggerPrg Struct Reference . . . . .	77
3.141 TriggerStep Struct Reference . . . . .	77
3.142Truck Class Reference . . . . .	78



3.142.1 Detailed Description . . . . .	79
3.142.2 Constructor & Destructor Documentation . . . . .	79
3.142.2.1 Truck(const double) . . . . .	79
3.142.3 Member Function Documentation . . . . .	79
3.142.3.1 addPackage(Package *) . . . . .	79
3.142.3.2 deliverPackage(Package *pack) . . . . .	79
3.142.3.3 getDirections() const . . . . .	80
3.142.3.4 getPackages() const . . . . .	80
3.142.3.5 getTruckID() const . . . . .	80
3.142.3.6 getWeight() const . . . . .	80
3.142.3.7 setDirections(vector< string * >) . . . . .	80
3.142.3.8 toString() const . . . . .	81
3.142.4 Member Data Documentation . . . . .	81
3.142.4.1 truckCount . . . . .	81
3.143unixFile Struct Reference . . . . .	81
3.144unixFileId Struct Reference . . . . .	82
3.145unixInodeInfo Struct Reference . . . . .	82
3.146unixShm Struct Reference . . . . .	82
3.147unixShmNode Struct Reference . . . . .	83
3.148UnixUnusedFd Struct Reference . . . . .	83
3.149UnpackedRecord Struct Reference . . . . .	83
3.150ValueNewStat4Ctx Struct Reference . . . . .	84
3.151Vdbe Struct Reference . . . . .	84
3.152VdbeCursor Struct Reference . . . . .	86
3.153VdbeFrame Struct Reference . . . . .	86
3.154VdbeOp Struct Reference . . . . .	87
3.155VdbeOpList Struct Reference . . . . .	88
3.156VdbeSorter Struct Reference . . . . .	88
3.157VtabCtx Struct Reference . . . . .	88
3.158VTable Struct Reference . . . . .	89

3.159vxworksFileId Struct Reference . . . . .	89
3.160Wal Struct Reference . . . . .	90
3.161WalCkptInfo Struct Reference . . . . .	90
3.162WalIndexHdr Struct Reference . . . . .	91
3.163WalIterator Struct Reference . . . . .	91
3.164Walker Struct Reference . . . . .	91
3.165WalIterator::WalSegment Struct Reference . . . . .	92
3.166WalWriter Struct Reference . . . . .	92
3.167WhereAndInfo Struct Reference . . . . .	93
3.168WhereClause Struct Reference . . . . .	93
3.169WhereInfo Struct Reference . . . . .	93
3.170WhereLevel Struct Reference . . . . .	94
3.171WhereLoop Struct Reference . . . . .	95
3.172WhereLoopBuilder Struct Reference . . . . .	96
3.173WhereMaskSet Struct Reference . . . . .	96
3.174WhereOrCost Struct Reference . . . . .	96
3.175WhereOrInfo Struct Reference . . . . .	97
3.176WhereOrSet Struct Reference . . . . .	97
3.177WherePath Struct Reference . . . . .	97
3.178WhereScan Struct Reference . . . . .	98
3.179WhereTerm Struct Reference . . . . .	98
3.180With Struct Reference . . . . .	99
3.181Parse::yColCache Struct Reference . . . . .	99
3.182YYMINORTYPE Union Reference . . . . .	99
3.183yyParser Struct Reference . . . . .	100
3.184yyStackEntry Struct Reference . . . . .	100

# Chapter 1

## CS3528\_Project

Final project for class CS3528

### Team Hope

- Eric Kuha
- Chelsey Riewer
- Robert Susmilch

### Basic Idea

To create a utility application for a (UPS) style package delivery system which operates in one town.

### Requirements

- Sort and order packages into truck
- Determine packages that can be delivered on time
  - Each truck can only operate for 8 hours per day
  - Trucks will have a weight limit
  - Assume it takes 1 minute to drive one block
  - Each stop takes 5 minutes
- Output should include list of packages for each truck and a set of driving directions for each driver.
- Packages have the following information
  - UUID
  - Two client objects: Sender and receiver
  - Priority: Overnight, Two Day, Regular
- Clients have the following information
  - UUID (Maybe?)
  - Name and address information

- A list record of sent packages
  - A list record of received packages
- The city has the following qualities
  - Assumed to be divided into quadrants
  - X axis is Main Street
  - Y axis is Central Avenue
  - Streets that are  $y=n$  are called nth Street North or nth Street South
  - Streets that are  $x=n$  are called nth Avenue East or nth Avenue West

## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Hash::_ht	7
AggInfo	7
AggInfo::AggInfo_col	7
AggInfo::AggInfo_func	8
analysisInfo	8
AuthContext	8
AutoincInfo	9
AuxData	9
Bitvec	9
BtCursor	10
BtLock	10
Btree	11
BtreePayload	11
BtShared	11
BusyHandler	12
CellArray	12
CellInfo	13
Client	
Client class	13
CollSeq	17
Column	17
compareInfo	18
CountCtx	18
With::Cte	18
DateTime	19
Db	19
DbFixer	19
DistinctCtx	20
et_info	20
Expr	20
ExprList	21
ExprList::ExprList_item	22
ExprSpan	22
FileChunk	22
FilePoint	23

<a href="#">FKey</a>	<a href="#">23</a>
<a href="#">fts5_api</a>	<a href="#">24</a>
<a href="#">fts5_tokenizer</a>	<a href="#">24</a>
<a href="#">Fts5ExtensionApi</a>	<a href="#">24</a>
<a href="#">Fts5Phraselter</a>	<a href="#">25</a>
<a href="#">FuncDef</a>	<a href="#">25</a>
<a href="#">FuncDefHash</a>	<a href="#">26</a>
<a href="#">FuncDestructor</a>	<a href="#">26</a>
<a href="#">Genetic</a>	<a href="#">26</a>
<a href="#">Hash</a>	<a href="#">27</a>
<a href="#">HashElem</a>	<a href="#">27</a>
<a href="#">IdList</a>	<a href="#">27</a>
<a href="#">IdList::IdList_item</a>	<a href="#">28</a>
<a href="#">IdxCover</a>	<a href="#">28</a>
<a href="#">ImportCtx</a>	<a href="#">28</a>
<a href="#">Incrblob</a>	<a href="#">29</a>
<a href="#">IncrMerger</a>	<a href="#">29</a>
<a href="#">Index</a>	<a href="#">29</a>
<a href="#">IndexSample</a>	<a href="#">30</a>
<a href="#">InitData</a>	<a href="#">30</a>
<a href="#">IntegrityCk</a>	<a href="#">31</a>
<a href="#">KeyInfo</a>	<a href="#">31</a>
<a href="#">LimitVal</a>	<a href="#">32</a>
<a href="#">Lookaside</a>	<a href="#">32</a>
<a href="#">LookasideSlot</a>	<a href="#">32</a>
<a href="#">Mem</a>	<a href="#">33</a>
<a href="#">MemJournal</a>	<a href="#">33</a>
<a href="#">MemPage</a>	<a href="#">34</a>
<a href="#">Mem::MemValue</a>	<a href="#">34</a>
<a href="#">MergeEngine</a>	<a href="#">35</a>
<a href="#">Module</a>	<a href="#">35</a>
<a href="#">NameContext</a>	<a href="#">35</a>
<a href="#">VdbeOp::p4union</a>	<a href="#">36</a>
<a href="#">Package</a>	
<a href="#">Package class</a>	<a href="#">36</a>
<a href="#">Pager</a>	<a href="#">40</a>
<a href="#">PagerSavepoint</a>	<a href="#">41</a>
<a href="#">Parse</a>	<a href="#">42</a>
<a href="#">PCache</a>	<a href="#">43</a>
<a href="#">PCache1</a>	<a href="#">44</a>
<a href="#">PgFreeslot</a>	<a href="#">44</a>
<a href="#">PgHdr</a>	<a href="#">44</a>
<a href="#">PgHdr1</a>	<a href="#">45</a>
<a href="#">PGroup</a>	<a href="#">45</a>
<a href="#">PmaReader</a>	<a href="#">46</a>
<a href="#">PmaWriter</a>	<a href="#">46</a>
<a href="#">PreUpdate</a>	<a href="#">47</a>
<a href="#">PrintfArguments</a>	<a href="#">47</a>
<a href="#">randomPackageEnum</a>	<a href="#">47</a>
<a href="#">ReusableSpace</a>	<a href="#">48</a>
<a href="#">RowSet</a>	<a href="#">48</a>
<a href="#">RowSetChunk</a>	<a href="#">48</a>
<a href="#">RowSetEntry</a>	<a href="#">49</a>
<a href="#">SavedModelInfo</a>	<a href="#">49</a>
<a href="#">Savepoint</a>	<a href="#">49</a>
<a href="#">ScanStatus</a>	<a href="#">50</a>
<a href="#">Schema</a>	<a href="#">50</a>
<a href="#">FKey::sColMap</a>	<a href="#">50</a>

ScratchFreeslot	51
Select	51
SelectDest	51
ShellState	52
SortCtx	53
SorterFile	53
SorterList	54
SorterRecord	54
SortSubtask	54
sqlite3	55
sqlite3_api_routines	56
sqlite3_backup	61
sqlite3_context	61
sqlite3_file	62
sqlite3_index_info::sqlite3_index_constraint	62
sqlite3_index_info::sqlite3_index_constraint_usage	62
sqlite3_index_info	62
sqlite3_index_info::sqlite3_index_orderby	63
sqlite3_io_methods	64
sqlite3_mem_methods	64
sqlite3_module	65
sqlite3_mutex	65
sqlite3_mutex_methods	65
sqlite3_pcache_methods	66
sqlite3_pcache_methods2	66
sqlite3_pcache_page	67
sqlite3_rtree_geometry	67
sqlite3_rtree_query_info	68
sqlite3_vfs	68
sqlite3_vtab	69
sqlite3_vtab_cursor	69
Sqlite3Config	69
sqlite3::sqlite3InitInfo	71
SQLiteThread	71
SrcCount	71
SrcList	72
SrcList::SrcList_item	72
Stat4Accum	73
Stat4Sample	73
StrAccum	74
SubProgram	74
SumCtx	74
Table	75
TableLock	75
TabResult	76
Token	76
TrigEvent	76
Trigger	76
TriggerPrg	77
TriggerStep	77
Truck	
Truck Class	78
unixFile	81
unixFileId	82
unixInodeInfo	82
unixShm	82
unixShmNode	83
UnixUnusedFd	83

<a href="#">UnpackedRecord</a>	83
<a href="#">ValueNewStat4Ctx</a>	84
<a href="#">Vdbe</a>	84
<a href="#">VdbeCursor</a>	86
<a href="#">VdbeFrame</a>	86
<a href="#">VdbeOp</a>	87
<a href="#">VdbeOpList</a>	88
<a href="#">VdbeSorter</a>	88
<a href="#">VtabCtx</a>	88
<a href="#">VTable</a>	89
<a href="#">vxworksField</a>	89
<a href="#">Wal</a>	90
<a href="#">WalCkptInfo</a>	90
<a href="#">WalIndexHdr</a>	91
<a href="#">WalIterator</a>	91
<a href="#">Walker</a>	91
<a href="#">WalIterator::WalSegment</a>	92
<a href="#">WalWriter</a>	92
<a href="#">WhereAndInfo</a>	93
<a href="#">WhereClause</a>	93
<a href="#">WhereInfo</a>	93
<a href="#">WhereLevel</a>	94
<a href="#">WhereLoop</a>	95
<a href="#">WhereLoopBuilder</a>	96
<a href="#">WhereMaskSet</a>	96
<a href="#">WhereOrCost</a>	96
<a href="#">WhereOrInfo</a>	97
<a href="#">WhereOrSet</a>	97
<a href="#">WherePath</a>	97
<a href="#">WhereScan</a>	98
<a href="#">WhereTerm</a>	98
<a href="#">With</a>	99
<a href="#">Parse::yColCache</a>	99
<a href="#">YYMINORTYPE</a>	99
<a href="#">yyParser</a>	100
<a href="#">yyStackEntry</a>	100



## Chapter 3

# Class Documentation

### 3.1 Hash::\_ht Struct Reference

Collaboration diagram for Hash::\_ht:

#### Public Attributes

- int **count**
- [HashElem](#) \* **chain**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.2 AggInfo Struct Reference

Collaboration diagram for AggInfo:

### 3.3 AggInfo::AggInfo\_col Struct Reference

Collaboration diagram for AggInfo::AggInfo\_col:

#### Public Attributes

- [Table](#) \* **pTab**
- int **iTable**
- int **iColumn**
- int **iSorterColumn**
- int **iMem**
- [Expr](#) \* **pExpr**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.4 AggInfo::AggInfo\_func Struct Reference

Collaboration diagram for AggInfo::AggInfo\_func:

#### Public Attributes

- [Expr](#) \* **pExpr**
- [FuncDef](#) \* **pFunc**
- int **iMem**
- int **iDistinct**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.5 analysisInfo Struct Reference

Collaboration diagram for analysisInfo:

#### Public Attributes

- [sqlite3](#) \* **db**
- const char \* **zDatabase**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.6 AuthContext Struct Reference

Collaboration diagram for AuthContext:

#### Public Attributes

- const char \* **zAuthContext**
- [Parse](#) \* **pParse**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.7 AutoincInfo Struct Reference

Collaboration diagram for AutoincInfo:

### Public Attributes

- [AutoincInfo](#) \* **pNext**
- [Table](#) \* **pTab**
- int **iDb**
- int **regCtr**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.8 AuxData Struct Reference

Collaboration diagram for AuxData:

### Public Attributes

- int **iOp**
- int **iArg**
- void \* **pAux**
- void(\* **xDelete** )(void \*)
- [AuxData](#) \* **pNext**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.9 Bitvec Struct Reference

Collaboration diagram for Bitvec:

### Public Attributes

- u32 **iSize**
- u32 **nSet**
- u32 **iDivisor**
- union {
  - BITVEC\_TELEM **aBitmap** [BITVEC\_NELEM]
  - u32 **aHash** [BITVEC\_NINT]
  - [Bitvec](#) \* **apSub** [BITVEC\_NPTR]
- } **u**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.10 BtCursor Struct Reference

Collaboration diagram for BtCursor:

#### Public Attributes

- [Btree](#) \* **pBtree**
- [BtShared](#) \* **pBt**
- [BtCursor](#) \* **pNext**
- Pgno \* **aOverflow**
- [CellInfo](#) **info**
- i64 **nKey**
- void \* **pKey**
- Pgno **pgnoRoot**
- int **nOvflAlloc**
- int **skipNext**
- u8 **curFlags**
- u8 **curPagerFlags**
- u8 **eState**
- u8 **hints**
- i8 **iPage**
- u8 **curlntKey**
- struct [KeyInfo](#) \* **pKeyInfo**
- void \* **padding1**
- u16 **aildx** [BTCURSOR\_MAX\_DEPTH]
- [MemPage](#) \* **apPage** [BTCURSOR\_MAX\_DEPTH]

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.11 BtLock Struct Reference

Collaboration diagram for BtLock:

#### Public Attributes

- [Btree](#) \* **pBtree**
- Pgno **iTable**
- u8 **eLock**
- [BtLock](#) \* **pNext**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.12 Btree Struct Reference

Collaboration diagram for Btree:

### Public Attributes

- [sqlite3](#) \* **db**
- [BtShared](#) \* **pBt**
- u8 **inTrans**
- u8 **sharable**
- u8 **locked**
- u8 **hasIncrblobCur**
- int **wantToLock**
- int **nBackup**
- u32 **iDataVersion**
- [Btree](#) \* **pNext**
- [Btree](#) \* **pPrev**
- [BtLock](#) **lock**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.13 BtreePayload Struct Reference

### Public Attributes

- const void \* **pKey**
- [sqlite3\\_int64](#) **nKey**
- const void \* **pData**
- int **nData**
- int **nZero**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.14 BtShared Struct Reference

Collaboration diagram for BtShared:

### Public Attributes

- [Pager](#) \* **pPager**
- [sqlite3](#) \* **db**
- [BtCursor](#) \* **pCursor**
- [MemPage](#) \* **pPage1**
- u8 **openFlags**
- u8 **autoVacuum**
- u8 **incrVacuum**
- u8 **bDoTruncate**
- u8 **inTransaction**
- u8 **max1bytePayload**
- u16 **btsFlags**
- u16 **maxLocal**
- u16 **minLocal**
- u16 **maxLeaf**
- u16 **minLeaf**
- u32 **pageSize**
- u32 **usableSize**
- int **nTransaction**
- u32 **nPage**
- void \* **pSchema**
- void(\* **xFreeSchema** )(void \*)
- [sqlite3\\_mutex](#) \* **mutex**
- [Bitvec](#) \* **pHasContent**
- int **nRef**
- [BtShared](#) \* **pNext**
- [BtLock](#) \* **pLock**
- [Btree](#) \* **pWriter**
- u8 \* **pTmpSpace**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.15 BusyHandler Struct Reference

### Public Attributes

- int(\* **xFunc** )(void \*, int)
- void \* **pArg**
- int **nBusy**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.16 CellArray Struct Reference

Collaboration diagram for CellArray:

### Public Attributes

- int **nCell**
- [MemPage](#) \* **pRef**
- u8 \*\* **apCell**
- u16 \* **szCell**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.17 CellInfo Struct Reference

### Public Attributes

- i64 **nKey**
- u8 \* **pPayload**
- u32 **nPayload**
- u16 **nLocal**
- u16 **nSize**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.18 Client Class Reference

[Client](#) class.

```
#include <Client.h>
```

### Public Member Functions

- [Client](#) (const string &nam, const string &add, const string &cit, const string &st, const string &z, const int &clientID)  
*[Client](#) basic constructor.*
- void [setName](#) (const string &)  
*Setter for name.*
- string [getName](#) () const  
*Getter for name.*
- void [setAddress](#) (const string &add, const string &cit, const string &st, const string &z)  
*Setter for address.*
- string [getAddress](#) () const  
*Getter for address.*
- string [getCity](#) () const  
*Getter for city.*
- string [getState](#) () const

- *Getter for state.*
- string `getZip ()` const
- *Getter for zip.*
- void `sendPackage (Package *pack)`
- *Add a sent package.*
- vector< `Package *` > `getSentPackages ()` const
- *Get sent packages.*
- void `receivePackage (Package *pack)`
- *Add a received package.*
- vector< `Package *` > `getReceivedPackages ()` const
- *Get received packages.*
- string `toString ()` const
- *To String.*
- pair< int, int > `getCoords ()`
- *Returns coordinate pair of client.*
- `Client *` `getPointer ()`
- unsigned int `getID ()`

## Private Member Functions

- void `parseAddress ()`
- *Parses address into coordinates.*

## Private Attributes

- string `name` = "TEST"
- *Client name.*
- string `address` = ""
- *Client street address.*
- string `city` = ""
- *Client city.*
- string `state` = ""
- *Client state.*
- string `zip` = ""
- *Client zip. Note as string due to possible upgrades to none numeric zips (eg, UK.)*
- vector< `Package *` > `sentPackages`
- *Vector of packages sent by this client.*
- vector< `Package *` > `receivedPackages`
- *Vector of packages received by this client.*
- pair< int, int > `coordinates`
- *Coordinate pair of address in the city.*
- int `ID`
- *ID of client.*

## Friends

- std::ostream & `operator<<` (std::ostream &, const `Client` &)
- *Insertion Operator override.*



### 3.18.1 Detailed Description

[Client](#) class.

This client class defines a citizen of the town who can send and receive packages. Each person has a name, address, and vectors representing records of all packages sent and received.

### 3.18.2 Constructor & Destructor Documentation

**3.18.2.1** `Client::Client ( const string & nam, const string & add, const string & cit, const string & st, const string & z, const int & clientID )`

[Client](#) basic constructor.

Basic constructor.

Parameters

<i>nam</i>	Name of client.
<i>add</i>	Street address for client.
<i>cit</i>	City of client
<i>st</i>	State of client
<i>z</i>	ZIP code of client. Note string usage for international packages.

### 3.18.3 Member Function Documentation

**3.18.3.1** `string Client::getAddress ( ) const [inline]`

Getter for address.

Returns

String of street address

**3.18.3.2** `pair<int, int> Client::getCoords ( ) [inline]`

Returns coordinate pair of client.

Returns

Coordinate pair of client on City grid.

**3.18.3.3** `vector< Package * > Client::getReceivedPackages ( ) const`

Get received packages.

Returns

Vector of received packages

#### 3.18.3.4 `vector< Package * > Client::getSentPackages ( ) const`

Get sent packages.

##### Returns

Vector of sent packages

#### 3.18.3.5 `void Client::parseAddress ( ) [private]`

Parses address into coordinates.

Parses the address stored in object variables to Cartesian coordinates. Main Street runs East/West at  $Y = 0$ . Central Avenue runs North/South at  $X = 0$ . All avenues are N/S, while all streets are E/W. Building numbers are modulus 100, and rounded up or down to the nearest integer.

#### 3.18.3.6 `void Client::receivePackage ( Package * pack )`

Add a received package.

##### Parameters

<i>pack</i>	pointer to package
-------------	--------------------

#### 3.18.3.7 `void Client::sendPackage ( Package * pack )`

Add a sent package.

##### Parameters

<i>pack</i>	pointer to package
-------------	--------------------

#### 3.18.3.8 `void Client::setAddress ( const string & add, const string & cit, const string & st, const string & z )`

Setter for address.

Updates address for client.

##### Parameters

<i>add</i>	Street address for client.
<i>cit</i>	City of client
<i>st</i>	State of client
<i>z</i>	ZIP code of client. Note string usage for international packages.

## 3.18.3.9 string Client::toString ( ) const

To String.

Converts client information to string for debugging output.

## Returns

String containing client info for std::cout

Prints out client's name and address and also all packages associated with the client. Insertion operator only prints name and address.

## Returns

string representation of client and all sent and received packages

The documentation for this class was generated from the following files:

- Client.h
- Client.cpp

## 3.19 CollSeq Struct Reference

## Public Attributes

- char \* **zName**
- u8 **enc**
- void \* **pUser**
- int(\* **xCmp** )(void \*, int, const void \*, int, const void \*)
- void(\* **xDel** )(void \*)

The documentation for this struct was generated from the following file:

- sqlite3.c

## 3.20 Column Struct Reference

Collaboration diagram for Column:

## Public Attributes

- char \* **zName**
- Expr \* **pDflt**
- char \* **zColl**
- u8 **notNull**
- char **affinity**
- u8 **szEst**
- u8 **colFlags**

The documentation for this struct was generated from the following file:

- sqlite3.c

### 3.21 compareInfo Struct Reference

#### Public Attributes

- u8 **matchAll**
- u8 **matchOne**
- u8 **matchSet**
- u8 **noCase**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.22 CountCtx Struct Reference

#### Public Attributes

- i64 **n**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.23 With::Cte Struct Reference

Collaboration diagram for With::Cte:

#### Public Attributes

- char \* **zName**
- [ExprList](#) \* **pCols**
- [Select](#) \* **pSelect**
- const char \* **zCteErr**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.24 DateTime Struct Reference

### Public Attributes

- sqlite3\_int64 **iJD**
- int **Y**
- int **M**
- int **D**
- int **h**
- int **m**
- int **tz**
- double **s**
- char **validYMD**
- char **validHMS**
- char **validJD**
- char **validTZ**
- char **tzSet**

The documentation for this struct was generated from the following file:

- sqlite3.c

## 3.25 Db Struct Reference

Collaboration diagram for Db:

### Public Attributes

- char \* **zDbSName**
- [Btree](#) \* **pBt**
- u8 **safety\_level**
- u8 **bSyncSet**
- [Schema](#) \* **pSchema**

The documentation for this struct was generated from the following file:

- sqlite3.c

## 3.26 DbFixer Struct Reference

Collaboration diagram for DbFixer:

### Public Attributes

- [Parse](#) \* **pParse**
- [Schema](#) \* **pSchema**
- int **bVarOnly**
- const char \* **zDb**
- const char \* **zType**
- const [Token](#) \* **pName**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.27 DistinctCtx Struct Reference

### Public Attributes

- u8 **isTnct**
- u8 **eTnctType**
- int **tabTnct**
- int **addrTnct**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.28 et\_info Struct Reference

### Public Attributes

- char **fmttype**
- etByte **base**
- etByte **flags**
- etByte **type**
- etByte **charset**
- etByte **prefix**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.29 Expr Struct Reference

Collaboration diagram for Expr:

### Public Attributes

- u8 **op**
- char **affinity**
- u32 **flags**
- union {
  - char \* **zToken**
  - int **iValue****u**
- [Expr](#) \* **pLeft**
- [Expr](#) \* **pRight**
- union {
  - [ExprList](#) \* **pList**
  - [Select](#) \* **pSelect****x**
- int **nHeight**
- int **iTable**
- ynVar **iColumn**
- i16 **iAgg**
- i16 **iRightJoinTable**
- u8 **op2**
- [AggInfo](#) \* **pAggInfo**
- [Table](#) \* **pTab**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.30 ExprList Struct Reference

Collaboration diagram for ExprList:

### Classes

- struct [ExprList\\_item](#)

### Public Attributes

- int **nExpr**
- struct [ExprList::ExprList\\_item](#) \* **a**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

### 3.31 ExprList::ExprList\_item Struct Reference

Collaboration diagram for ExprList::ExprList\_item:

#### Public Attributes

- [Expr](#) \* **pExpr**
- char \* **zName**
- char \* **zSpan**
- u8 **sortOrder**
- unsigned **done**:1
- unsigned **bSpanIsTab**:1
- unsigned **reusable**:1
- union {  
    struct {  
        u16 **iOrderByCol**  
        u16 **iAlias**  
    } **x**  
    int **iConstExprReg**  
} **u**

The documentation for this struct was generated from the following file:

- sqlite3.c

### 3.32 ExprSpan Struct Reference

Collaboration diagram for ExprSpan:

#### Public Attributes

- [Expr](#) \* **pExpr**
- const char \* **zStart**
- const char \* **zEnd**

The documentation for this struct was generated from the following file:

- sqlite3.c

### 3.33 FileChunk Struct Reference

Collaboration diagram for FileChunk:



### Public Attributes

- [FileChunk](#) \* **pNext**
- u8 **zChunk** [8]

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.34 FilePoint Struct Reference

Collaboration diagram for FilePoint:

### Public Attributes

- `sqlite3_int64` **iOffset**
- [FileChunk](#) \* **pChunk**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.35 FKey Struct Reference

Collaboration diagram for FKey:

### Classes

- struct [sColMap](#)

### Public Attributes

- [Table](#) \* **pFrom**
- [FKey](#) \* **pNextFrom**
- char \* **zTo**
- [FKey](#) \* **pNextTo**
- [FKey](#) \* **pPrevTo**
- int **nCol**
- u8 **isDeferred**
- u8 **aAction** [2]
- [Trigger](#) \* **apTrigger** [2]
- struct [FKey::sColMap](#) **aCol** [1]

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.36 fts5\_api Struct Reference

#### Public Attributes

- int **iVersion**
- int(\* **xCreateTokenizer** )(fts5\_api \*pApi, const char \*zName, void \*pContext, fts5\_tokenizer \*pTokenizer, void(\*xDestroy)(void \*))
- int(\* **xFindTokenizer** )(fts5\_api \*pApi, const char \*zName, void \*\*ppContext, fts5\_tokenizer \*pTokenizer)
- int(\* **xCreateFunction** )(fts5\_api \*pApi, const char \*zName, void \*pContext, fts5\_extension\_function x↔Function, void(\*xDestroy)(void \*))

The documentation for this struct was generated from the following files:

- sqlite3.c
- sqlite3.h

### 3.37 fts5\_tokenizer Struct Reference

#### Public Attributes

- int(\* **xCreate** )(void \*, const char \*\*azArg, int nArg, Fts5Tokenizer \*\*ppOut)
- void(\* **xDelete** )(Fts5Tokenizer \*)
- int(\* **xTokenize** )(Fts5Tokenizer \*, void \*pCtx, int flags, const char \*pText, int nText, int(\*xToken)(void \*pCtx, int tflags, const char \*pToken, int nToken, int iStart, int iEnd))

The documentation for this struct was generated from the following files:

- sqlite3.c
- sqlite3.h

### 3.38 Fts5ExtensionApi Struct Reference

#### Public Attributes

- int **iVersion**
- void(\* **xUserData** )(Fts5Context \*)
- int(\* **xColumnCount** )(Fts5Context \*)
- int(\* **xRowCount** )(Fts5Context \*, sqlite3\_int64 \*pnRow)
- int(\* **xColumnTotalSize** )(Fts5Context \*, int iCol, sqlite3\_int64 \*pnToken)
- int(\* **xTokenize** )(Fts5Context \*, const char \*pText, int nText, void \*pCtx, int(\*xToken)(void \*, int, const char \*, int, int, int))
- int(\* **xPhraseCount** )(Fts5Context \*)
- int(\* **xPhraseSize** )(Fts5Context \*, int iPhrase)
- int(\* **xInstCount** )(Fts5Context \*, int \*pnInst)
- int(\* **xInst** )(Fts5Context \*, int iIdx, int \*piPhrase, int \*piCol, int \*piOff)
- sqlite3\_int64(\* **xRowid** )(Fts5Context \*)
- int(\* **xColumnText** )(Fts5Context \*, int iCol, const char \*\*pz, int \*pn)
- int(\* **xColumnSize** )(Fts5Context \*, int iCol, int \*pnToken)

- `int(* xQueryPhrase )(Fts5Context *, int iPhrase, void *pUserData, int*)(const Fts5ExtensionApi *, Fts5Context *, void *)`
- `int(* xSetAuxdata )(Fts5Context *, void *pAux, void(*xDelete)(void *))`
- `void *(* xGetAuxdata )(Fts5Context *, int bClear)`
- `int(* xPhraseFirst )(Fts5Context *, int iPhrase, Fts5Phraselter *, int *, int *)`
- `void(* xPhraseNext )(Fts5Context *, Fts5Phraselter *, int *piCol, int *piOff)`
- `int(* xPhraseFirstColumn )(Fts5Context *, int iPhrase, Fts5Phraselter *, int *)`
- `void(* xPhraseNextColumn )(Fts5Context *, Fts5Phraselter *, int *piCol)`

The documentation for this struct was generated from the following files:

- `sqlite3.c`
- `sqlite3.h`

### 3.39 Fts5Phraselter Struct Reference

#### Public Attributes

- `const unsigned char * a`
- `const unsigned char * b`

The documentation for this struct was generated from the following files:

- `sqlite3.c`
- `sqlite3.h`

### 3.40 FuncDef Struct Reference

Collaboration diagram for FuncDef:

#### Public Attributes

- `i8 nArg`
- `u16 funcFlags`
- `void * pUserData`
- `FuncDef * pNext`
- `void(* xSFunc )(sqlite3_context *, int, sqlite3_value **)`
- `void(* xFinalize )(sqlite3_context *)`
- `const char * zName`
- `union {  
    FuncDef * pHash  
    FuncDestructor * pDestructor  
} u`

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.41 FuncDefHash Struct Reference

Collaboration diagram for FuncDefHash:

#### Public Attributes

- [FuncDef](#) \* **a** [SQLITE\_FUNC\_HASH\_SZ]

The documentation for this struct was generated from the following file:

- sqlite3.c

### 3.42 FuncDestructor Struct Reference

#### Public Attributes

- int **nRef**
- void(\* **xDestroy** )(void \*)
- void \* **pUserData**

The documentation for this struct was generated from the following file:

- sqlite3.c

### 3.43 Genetic Class Reference

#### Public Member Functions

- [Genetic](#) (vector< [Package](#) \* > \*packs, vector< vector< unsigned int > > \*matrix, unsigned int weight, unsigned int packLimit, unsigned int population, float stops, float drive)  
*Default constructor \*/.*
- virtual [~Genetic](#) ()  
*Default destructor \*/.*
- void **initPopulation** ()
- float **fitness** (vector< [Client](#) \* >)

#### Private Attributes

- vector< [Package](#) \* > \* **packages**
- vector< vector< unsigned int > > \* **adMatrix**
- unsigned int **popNum**
- unsigned int **weightLimit**
- unsigned int **packageLimit**
- float **stopTime**
- float **driveTime**
- vector< vector< [Client](#) \* > > **genes**
- vector< [Client](#) \* > **bestFit**

The documentation for this class was generated from the following files:

- Genetic.h
- Genetic.cpp

## 3.44 Hash Struct Reference

Collaboration diagram for Hash:

### Classes

- struct [\\_ht](#)

### Public Attributes

- unsigned int **htsize**
- unsigned int **count**
- [HashElem](#) \* **first**
- struct [Hash::\\_ht](#) \* **ht**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.45 HashElem Struct Reference

Collaboration diagram for HashElem:

### Public Attributes

- [HashElem](#) \* **next**
- [HashElem](#) \* **prev**
- void \* **data**
- const char \* **pKey**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.46 IdList Struct Reference

Collaboration diagram for IdList:

### Classes

- struct [IdList\\_item](#)

### Public Attributes

- struct [IdList::IdList\\_item](#) \* **a**
- int **nId**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.47 IdList::IdList\_item Struct Reference

### Public Attributes

- char \* **zName**
- int **idx**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.48 IdxCover Struct Reference

Collaboration diagram for IdxCover:

### Public Attributes

- [Index](#) \* **pIdx**
- int **iCur**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.49 ImportCtx Struct Reference

### Public Attributes

- const char \* **zFile**
- FILE \* **in**
- char \* **z**
- int **n**
- int **nAlloc**
- int **nLine**
- int **cTerm**
- int **cColSep**
- int **cRowSep**

The documentation for this struct was generated from the following file:

- [shell.c](#)

## 3.50 Incrblob Struct Reference

Collaboration diagram for Incrblob:

### Public Attributes

- int **flags**
- int **nByte**
- int **iOffset**
- int **iCol**
- [BtCursor](#) \* **pCsr**
- sqlite3\_stmt \* **pStmt**
- [sqlite3](#) \* **db**
- char \* **zDb**
- [Table](#) \* **pTab**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.51 IncrMerger Struct Reference

Collaboration diagram for IncrMerger:

### Public Attributes

- [SortSubtask](#) \* **pTask**
- [MergeEngine](#) \* **pMerger**
- i64 **iStartOff**
- int **mxSz**
- int **bEof**
- int **bUseThread**
- [SorterFile](#) **aFile** [2]

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.52 Index Struct Reference

Collaboration diagram for Index:

## Public Attributes

- char \* **zName**
- i16 \* **aiColumn**
- LogEst \* **aiRowLogEst**
- [Table](#) \* **pTable**
- char \* **zColAff**
- [Index](#) \* **pNext**
- [Schema](#) \* **pSchema**
- u8 \* **aSortOrder**
- const char \*\* **azColl**
- [Expr](#) \* **pPartIdxWhere**
- [ExprList](#) \* **aColExpr**
- int **tnum**
- LogEst **szIdxRow**
- u16 **nKeyCol**
- u16 **nColumn**
- u8 **onError**
- unsigned **idxType**:2
- unsigned **bUnordered**:1
- unsigned **uniqNotNull**:1
- unsigned **isResized**:1
- unsigned **isCovering**:1
- unsigned **noSkipScan**:1

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.53 IndexSample Struct Reference

### Public Attributes

- void \* **p**
- int **n**
- tRowcnt \* **anEq**
- tRowcnt \* **anLt**
- tRowcnt \* **anDLt**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.54 InitData Struct Reference

Collaboration diagram for InitData:



### Public Attributes

- [sqlite3](#) \* **db**
- char \*\* **pzErrMsg**
- int **iDb**
- int **rc**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.55 IntegrityCk Struct Reference

Collaboration diagram for IntegrityCk:

### Public Attributes

- [BtShared](#) \* **pBt**
- [Pager](#) \* **pPager**
- u8 \* **aPgRef**
- Pgno **nPage**
- int **mxErr**
- int **nErr**
- int **mallocFailed**
- const char \* **zPfx**
- int **v1**
- int **v2**
- [StrAccum](#) **errMsg**
- u32 \* **heap**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.56 KeyInfo Struct Reference

Collaboration diagram for KeyInfo:

### Public Attributes

- u32 **nRef**
- u8 **enc**
- u16 **nField**
- u16 **nXField**
- [sqlite3](#) \* **db**
- u8 \* **aSortOrder**
- [CollSeq](#) \* **aColl** [1]

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

### 3.57 LimitVal Struct Reference

Collaboration diagram for LimitVal:

#### Public Attributes

- [Expr](#) \* **pLimit**
- [Expr](#) \* **pOffset**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.58 Lookaside Struct Reference

Collaboration diagram for Lookaside:

#### Public Attributes

- u32 **bDisable**
- u16 **sz**
- u8 **bMalloced**
- int **nOut**
- int **mxOut**
- int **anStat** [3]
- [LookasideSlot](#) \* **pFree**
- void \* **pStart**
- void \* **pEnd**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.59 LookasideSlot Struct Reference

Collaboration diagram for LookasideSlot:

#### Public Attributes

- [LookasideSlot](#) \* **pNext**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.60 Mem Struct Reference

Collaboration diagram for Mem:

### Classes

- union [MemValue](#)

### Public Attributes

- union [Mem::MemValue](#) **u**
- u16 **flags**
- u8 **enc**
- u8 **eSubtype**
- int **n**
- char \* **z**
- char \* **zMalloc**
- int **szMalloc**
- u32 **uTemp**
- [sqlite3](#) \* **db**
- void(\* **xDel** )(void \*)

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.61 MemJournal Struct Reference

Collaboration diagram for MemJournal:

### Public Attributes

- const [sqlite3\\_io\\_methods](#) \* **pMethod**
- int **nChunkSize**
- int **nSpill**
- int **nSize**
- [FileChunk](#) \* **pFirst**
- [FilePoint](#) **endpoint**
- [FilePoint](#) **readpoint**
- int **flags**
- [sqlite3\\_vfs](#) \* **pVfs**
- const char \* **zJournal**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

### 3.62 MemPage Struct Reference

Collaboration diagram for MemPage:

#### Public Attributes

- u8 **isInit**
- u8 **nOverflow**
- u8 **intKey**
- u8 **intKeyLeaf**
- u8 **leaf**
- u8 **hdrOffset**
- u8 **childPtrSize**
- u8 **max1bytePayload**
- u8 **bBusy**
- u16 **maxLocal**
- u16 **minLocal**
- u16 **cellOffset**
- u16 **nFree**
- u16 **nCell**
- u16 **maskPage**
- u16 **aiOvfl** [5]
- u8 \* **apOvfl** [5]
- [BtShared](#) \* **pBt**
- u8 \* **aData**
- u8 \* **aDataEnd**
- u8 \* **aCellIdx**
- u8 \* **aDataOfst**
- [DbPage](#) \* **pDbPage**
- u16(\* **xCellSize** )(MemPage \*, u8 \*)
- void(\* **xParseCell** )(MemPage \*, u8 \*, [CellInfo](#) \*)
- Pgno **pgno**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

### 3.63 Mem::MemValue Union Reference

Collaboration diagram for Mem::MemValue:

#### Public Attributes

- double **r**
- i64 **i**
- int **nZero**
- [FuncDef](#) \* **pDef**
- [RowSet](#) \* **pRowSet**
- [VdbeFrame](#) \* **pFrame**

The documentation for this union was generated from the following file:

- [sqlite3.c](#)

## 3.64 MergeEngine Struct Reference

Collaboration diagram for MergeEngine:

### Public Attributes

- int **nTree**
- [SortSubtask](#) \* **pTask**
- int \* **aTree**
- [PmaReader](#) \* **aReadr**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.65 Module Struct Reference

Collaboration diagram for Module:

### Public Attributes

- const [sqlite3\\_module](#) \* **pModule**
- const char \* **zName**
- void \* **pAux**
- void(\* **xDestroy** )(void \*)
- [Table](#) \* **pEpoTab**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.66 NameContext Struct Reference

Collaboration diagram for NameContext:

### Public Attributes

- [Parse](#) \* **pParse**
- [SrcList](#) \* **pSrcList**
- [ExprList](#) \* **pEList**
- [AgglInfo](#) \* **pAgglInfo**
- [NameContext](#) \* **pNext**
- int **nRef**
- int **nErr**
- u16 **ncFlags**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.67 VdbeOp::p4union Union Reference

Collaboration diagram for VdbeOp::p4union:

#### Public Attributes

- int **i**
- void \* **p**
- char \* **z**
- i64 \* **pi64**
- double \* **pReal**
- [FuncDef](#) \* **pFunc**
- [sqlite3\\_context](#) \* **pCtx**
- [CollSeq](#) \* **pColl**
- [Mem](#) \* **pMem**
- [VTable](#) \* **pVtab**
- [KeyInfo](#) \* **pKeyInfo**
- int \* **ai**
- [SubProgram](#) \* **pProgram**
- [Table](#) \* **pTab**
- int(\* **xAdvance** )([BtCursor](#) \*, int \*)

The documentation for this union was generated from the following file:

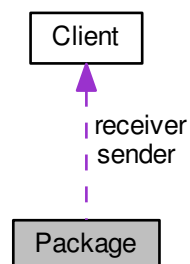
- [sqlite3.c](#)

### 3.68 Package Class Reference

[Package](#) class.

```
#include <Package.h>
```

Collaboration diagram for Package:



## Public Member Functions

- [Package](#) ([Client](#) \*s, [Client](#) \*r, const float w, const Priority p)  
*Constructor.*
- void [setSender](#) ([Client](#) \*)  
*Sender setter.*
- [Client](#) \* [getSender](#) () const  
*Sender getter.*
- void [setReceiver](#) ([Client](#) \*)  
*Receiver setter.*
- [Client](#) \* [getReceiver](#) () const  
*Receiver getter.*
- void [setWeight](#) (const double)  
*Weight setter.*
- float [getWeight](#) () const  
*Weight getter.*
- void [setPriority](#) (const Priority)  
*Priority setter.*
- Priority [getPriority](#) () const  
*Priority getter.*
- string [getPriorityString](#) () const  
*Priority string getter.*
- [Package](#) \* [getPointer](#) ()

## Private Attributes

- [Client](#) \* [sender](#)  
*Pointer to sender client.*
- [Client](#) \* [receiver](#)  
*Pointer to receiver client.*
- float [weight](#)  
*Weight of package in ounces.*
- Priority [priority](#)  
*Priority enum of package.*
- bool [Delivered](#) = false  
*Package delivered?*

## Friends

- std::ostream & [operator<<](#) (std::ostream &, const [Package](#) &)  
*Insertion Operator Override.*

### 3.68.1 Detailed Description

[Package](#) class.

This [Package](#) class defines a package which was sent by a [Client](#) and received by a [Client](#). Each package has a priority, a weight, and two clients (a sender and receiver).

### 3.68.2 Constructor & Destructor Documentation

#### 3.68.2.1 [Package::Package](#) ( [Client](#) \* s, [Client](#) \* r, const float w, const Priority p )

Constructor.

## Parameters

<i>s</i>	Sender <a href="#">Client</a>
<i>r</i>	Receiver <a href="#">Client</a>
<i>w</i>	<a href="#">Package</a> weight
<i>p</i>	<a href="#">Package</a> priority

### 3.68.3 Member Function Documentation

#### 3.68.3.1 `Priority Package::getPriority ( ) const`

Priority getter.

[Package](#) priority getter.

## Returns

[Package](#) priority

#### 3.68.3.2 `string Package::getPriorityString ( ) const`

Priority string getter.

Priority is an enum and hence is actually an int. Here's a nice way to print the string value.

## Returns

string of priority

#### 3.68.3.3 `Client * Package::getReceiver ( ) const`

Receiver getter.

## Returns

pointer to receiver client

#### 3.68.3.4 `Client * Package::getSender ( ) const`

Sender getter.

## Returns

pointer to sender client



#### 3.68.3.5 float Package::getWeight ( ) const

Weight getter.

[Package](#) Weight getter.

Returns

[Package](#) weight

#### 3.68.3.6 void Package::setPriority ( const Priority *p* )

Priority setter.

[Package](#) Priority setter.

Parameters

<i>p</i>	Priority enum
----------	---------------

#### 3.68.3.7 void Package::setReceiver ( Client \* *r* )

Receiver setter.

Parameters

<i>r</i>	pointer to receiver client
----------	----------------------------

#### 3.68.3.8 void Package::setSender ( Client \* *s* )

Sender setter.

Parameters

<i>s</i>	pointer to sender client
----------	--------------------------

#### 3.68.3.9 void Package::setWeight ( const double *w* )

Weight setter.

[Package](#) weight setter.

Throws error on negative weight.

## Parameters

<i>w</i>	<a href="#">Package</a> weight
----------	--------------------------------

The documentation for this class was generated from the following files:

- [Package.h](#)
- [Package.cpp](#)

### 3.69 Pager Struct Reference

Collaboration diagram for Pager:

#### Public Attributes

- [sqlite3\\_vfs](#) \* **pVfs**
- u8 **exclusiveMode**
- u8 **journalMode**
- u8 **useJournal**
- u8 **noSync**
- u8 **fullSync**
- u8 **extraSync**
- u8 **ckptSyncFlags**
- u8 **walSyncFlags**
- u8 **syncFlags**
- u8 **tempFile**
- u8 **noLock**
- u8 **readOnly**
- u8 **memDb**
- u8 **eState**
- u8 **eLock**
- u8 **changeCountDone**
- u8 **setMaster**
- u8 **doNotSpill**
- u8 **subjInMemory**
- u8 **bUseFetch**
- u8 **hasHeldSharedLock**
- Pgno **dbSize**
- Pgno **dbOrigSize**
- Pgno **dbFileSize**
- Pgno **dbHintSize**
- int **errCode**
- int **nRec**
- u32 **cksumInit**
- u32 **nSubRec**
- [Bitvec](#) \* **plnJournal**
- [sqlite3\\_file](#) \* **fd**
- [sqlite3\\_file](#) \* **jfd**
- [sqlite3\\_file](#) \* **sjfd**
- i64 **journalOff**

- i64 **journalHdr**
- [sqlite3\\_backup](#) \* **pBackup**
- [PagerSavepoint](#) \* **aSavepoint**
- int **nSavepoint**
- u32 **iDataVersion**
- char **dbFileVers** [16]
- int **nMmapOut**
- sqlite3\_int64 **szMmap**
- [PgHdr](#) \* **pMmapFreelist**
- u16 **nExtra**
- i16 **nReserve**
- u32 **vfsFlags**
- u32 **sectorSize**
- int **pageSize**
- Pgno **mxPgno**
- i64 **journalSizeLimit**
- char \* **zFilename**
- char \* **zJournal**
- int(\* **xBusyHandler** )(void \*)
- void \* **pBusyHandlerArg**
- int **aStat** [3]
- void(\* **xReinit** )(DbPage \*)
- char \* **pTmpSpace**
- [PCache](#) \* **pPCache**
- [Wal](#) \* **pWal**
- char \* **zWal**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.70 PagerSavepoint Struct Reference

Collaboration diagram for PagerSavepoint:

### Public Attributes

- i64 **iOffset**
- i64 **iHdrOffset**
- [Bitvec](#) \* **pInSavepoint**
- Pgno **nOrig**
- Pgno **iSubRec**
- u32 **aWalData** [WAL\_SAVEPOINT\_NDATA]

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

### 3.71 Parse Struct Reference

Collaboration diagram for Parse:

#### Classes

- struct [yColCache](#)

#### Public Attributes

- [sqlite3](#) \* **db**
- char \* **zErrMsg**
- [Vdbe](#) \* **pVdbe**
- int **rc**
- u8 **colNamesSet**
- u8 **checkSchema**
- u8 **nested**
- u8 **nTempReg**
- u8 **isMultiWrite**
- u8 **mayAbort**
- u8 **hasCompound**
- u8 **okConstFactor**
- u8 **disableLookaside**
- u8 **nColCache**
- int **nRangeReg**
- int **iRangeReg**
- int **nErr**
- int **nTab**
- int **nMem**
- int **nOpAlloc**
- int **szOpAlloc**
- int **ckBase**
- int **iSelfTab**
- int **iCacheLevel**
- int **iCacheCnt**
- int **nLabel**
- int \* **aLabel**
- [ExprList](#) \* **pConstExpr**
- [Token](#) **constraintName**
- yDbMask **writeMask**
- yDbMask **cookieMask**
- int **regRowid**
- int **regRoot**
- int **nMaxArg**
- int **nTableLock**
- [TableLock](#) \* **aTableLock**
- [AutoincInfo](#) \* **pAinc**
- [Parse](#) \* **pToplevel**
- [Table](#) \* **pTriggerTab**
- int **addrCrTab**
- u32 **nQueryLoop**
- u32 **oldmask**

- u32 **newmask**
- u8 **eTriggerOp**
- u8 **eOrconf**
- u8 **disableTriggers**
- struct [Parse::yColCache](#) **aColCache** [SQLITE\_N\_COLCACHE]
- int **aTempReg** [8]
- [Token](#) **sNameToken**
- [Token](#) **sLastToken**
- ynVar **nVar**
- int **nzVar**
- u8 **iPkSortOrder**
- u8 **explain**
- u8 **declareVtab**
- int **nVtabLock**
- int **nHeight**
- int **iSelectId**
- int **iNextSelectId**
- char \*\* **azVar**
- [Vdbe](#) \* **pReprepare**
- const char \* **zTail**
- [Table](#) \* **pNewTable**
- [Trigger](#) \* **pNewTrigger**
- const char \* **zAuthContext**
- [Token](#) **sArg**
- [Table](#) \*\* **apVtabLock**
- [Table](#) \* **pZombieTab**
- [TriggerPrg](#) \* **pTriggerPrg**
- [With](#) \* **pWith**
- [With](#) \* **pWithToFree**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.72 PCache Struct Reference

Collaboration diagram for PCache:

### Public Attributes

- [PgHdr](#) \* **pDirty**
- [PgHdr](#) \* **pDirtyTail**
- [PgHdr](#) \* **pSynced**
- int **nRefSum**
- int **szCache**
- int **szSpill**
- int **szPage**
- int **szExtra**
- u8 **bPurgeable**
- u8 **eCreate**
- int(\* **xStress** )(void \*, [PgHdr](#) \*)
- void \* **pStress**
- `sqlite3_pcache` \* **pCache**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.73 PCache1 Struct Reference

Collaboration diagram for PCache1:

#### Public Attributes

- [PGroup](#) \* **pGroup**
- int **szPage**
- int **szExtra**
- int **szAlloc**
- int **bPurgeable**
- unsigned int **nMin**
- unsigned int **nMax**
- unsigned int **n90pct**
- unsigned int **iMaxKey**
- unsigned int **nRecyclable**
- unsigned int **nPage**
- unsigned int **nHash**
- [PgHdr1](#) \*\* **apHash**
- [PgHdr1](#) \* **pFree**
- void \* **pBulk**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.74 PgFreeslot Struct Reference

Collaboration diagram for PgFreeslot:

#### Public Attributes

- [PgFreeslot](#) \* **pNext**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.75 PgHdr Struct Reference

Collaboration diagram for PgHdr:

### Public Attributes

- [sqlite3\\_pcache\\_page](#) \* **pPage**
- void \* **pData**
- void \* **pExtra**
- [PgHdr](#) \* **pDirty**
- [Pager](#) \* **pPager**
- Pgno **pgno**
- u16 **flags**
- i16 **nRef**
- [PCache](#) \* **pCache**
- [PgHdr](#) \* **pDirtyNext**
- [PgHdr](#) \* **pDirtyPrev**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.76 PgHdr1 Struct Reference

Collaboration diagram for PgHdr1:

### Public Attributes

- [sqlite3\\_pcache\\_page](#) **page**
- unsigned int **iKey**
- u8 **isPinned**
- u8 **isBulkLocal**
- u8 **isAnchor**
- [PgHdr1](#) \* **pNext**
- [PCache1](#) \* **pCache**
- [PgHdr1](#) \* **pLruNext**
- [PgHdr1](#) \* **pLruPrev**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.77 PGroup Struct Reference

Collaboration diagram for PGroup:

### Public Attributes

- [sqlite3\\_mutex](#) \* **mutex**
- unsigned int **nMaxPage**
- unsigned int **nMinPage**
- unsigned int **mxPinned**
- unsigned int **nCurrentPage**
- [PgHdr1](#) **lru**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.78 PmaReader Struct Reference

Collaboration diagram for PmaReader:

### Public Attributes

- i64 **iReadOff**
- i64 **iEof**
- int **nAlloc**
- int **nKey**
- [sqlite3\\_file](#) \* **pFd**
- u8 \* **aAlloc**
- u8 \* **aKey**
- u8 \* **aBuffer**
- int **nBuffer**
- u8 \* **aMap**
- [IncrMerger](#) \* **pIncr**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.79 PmaWriter Struct Reference

Collaboration diagram for PmaWriter:

### Public Attributes

- int **eFWErr**
- u8 \* **aBuffer**
- int **nBuffer**
- int **iBufStart**
- int **iBufEnd**
- i64 **iWriteOff**
- [sqlite3\\_file](#) \* **pFd**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)



## 3.80 PreUpdate Struct Reference

Collaboration diagram for PreUpdate:

### Public Attributes

- [Vdbe](#) \* **v**
- [VdbeCursor](#) \* **pCsr**
- int **op**
- u8 \* **aRecord**
- [KeyInfo](#) **keyinfo**
- [UnpackedRecord](#) \* **pUnpacked**
- [UnpackedRecord](#) \* **pNewUnpacked**
- int **iNewReg**
- i64 **iKey1**
- i64 **iKey2**
- [Mem](#) \* **aNew**
- [Table](#) \* **pTab**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.81 PrintfArguments Struct Reference

Collaboration diagram for PrintfArguments:

### Public Attributes

- int **nArg**
- int **nUsed**
- [sqlite3\\_value](#) \*\* **apArg**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.82 randomPackageEnum Struct Reference

### Public Attributes

- string **fileName**
- unsigned int **population**
- unsigned int **num**
- int **maxAddress**
- unsigned int **maxStreets**
- float **maxWeight**
- float **priority** [3]

The documentation for this struct was generated from the following file:

- `client_test.cpp`

### 3.83 ReusableSpace Struct Reference

#### Public Attributes

- u8 \* **pSpace**
- int **nFree**
- int **nNeeded**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.84 RowSet Struct Reference

Collaboration diagram for RowSet:

#### Public Attributes

- struct [RowSetChunk](#) \* **pChunk**
- [sqlite3](#) \* **db**
- struct [RowSetEntry](#) \* **pEntry**
- struct [RowSetEntry](#) \* **pLast**
- struct [RowSetEntry](#) \* **pFresh**
- struct [RowSetEntry](#) \* **pForest**
- u16 **nFresh**
- u16 **rsFlags**
- int **iBatch**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.85 RowSetChunk Struct Reference

Collaboration diagram for RowSetChunk:

#### Public Attributes

- struct [RowSetChunk](#) \* **pNextChunk**
- struct [RowSetEntry](#) **aEntry** [ROWSET\_ENTRY\_PER\_CHUNK]

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.86 RowSetEntry Struct Reference

Collaboration diagram for RowSetEntry:

### Public Attributes

- i64 **v**
- struct RowSetEntry \* **pRight**
- struct RowSetEntry \* **pLeft**

The documentation for this struct was generated from the following file:

- sqlite3.c

## 3.87 SavedModelInfo Struct Reference

### Public Attributes

- int **valid**
- int **mode**
- int **showHeader**
- int **colWidth** [100]

The documentation for this struct was generated from the following file:

- shell.c

## 3.88 Savepoint Struct Reference

Collaboration diagram for Savepoint:

### Public Attributes

- char \* **zName**
- i64 **nDeferredCons**
- i64 **nDeferredImmCons**
- Savepoint \* **pNext**

The documentation for this struct was generated from the following file:

- sqlite3.c

### 3.89 ScanStatus Struct Reference

#### Public Attributes

- int **addrExplain**
- int **addrLoop**
- int **addrVisit**
- int **iSelectID**
- LogEst **nEst**
- char \* **zName**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.90 Schema Struct Reference

Collaboration diagram for Schema:

#### Public Attributes

- int **schema\_cookie**
- int **iGeneration**
- Hash **tblHash**
- Hash **idxHash**
- Hash **trigHash**
- Hash **fkeyHash**
- Table \* **pSeqTab**
- u8 **file\_format**
- u8 **enc**
- u16 **schemaFlags**
- int **cache\_size**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.91 FKey::sColMap Struct Reference

#### Public Attributes

- int **iFrom**
- char \* **zCol**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.92 ScratchFreeslot Struct Reference

Collaboration diagram for ScratchFreeslot:

### Public Attributes

- struct [ScratchFreeslot](#) \* **pNext**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.93 Select Struct Reference

Collaboration diagram for Select:

### Public Attributes

- [ExprList](#) \* **pEList**
- u8 **op**
- LogEst **nSelectRow**
- u32 **selFlags**
- int **iLimit**
- int **iOffset**
- int **addrOpenEphm** [2]
- [SrcList](#) \* **pSrc**
- [Expr](#) \* **pWhere**
- [ExprList](#) \* **pGroupBy**
- [Expr](#) \* **pHaving**
- [ExprList](#) \* **pOrderBy**
- [Select](#) \* **pPrior**
- [Select](#) \* **pNext**
- [Expr](#) \* **pLimit**
- [Expr](#) \* **pOffset**
- [With](#) \* **pWith**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.94 SelectDest Struct Reference

Collaboration diagram for SelectDest:

## Public Attributes

- u8 **eDest**
- char \* **zAffSdst**
- int **iSDParm**
- int **iSdst**
- int **nSdst**
- [ExprList](#) \* **pOrderBy**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.95 ShellState Struct Reference

Collaboration diagram for ShellState:

## Public Attributes

- [sqlite3](#) \* **db**
- int **echoOn**
- int **autoExplain**
- int **autoEQP**
- int **statsOn**
- int **scanstatsOn**
- int **countChanges**
- int **backslashOn**
- int **outCount**
- int **cnt**
- FILE \* **out**
- FILE \* **traceOut**
- int **nErr**
- int **mode**
- int **cMode**
- int **normalMode**
- int **writableSchema**
- int **showHeader**
- int **nCheck**
- unsigned **shellFlgs**
- char \* **zDestTable**
- char **zTestcase** [30]
- char **colSeparator** [20]
- char **rowSeparator** [20]
- int **colWidth** [100]
- int **actualWidth** [100]
- char **nullValue** [20]
- char **outfile** [FILENAME\_MAX]
- const char \* **zDbFilename**
- char \* **zFreeOnClose**
- const char \* **zVfs**
- [sqlite3\\_stmt](#) \* **pStmt**

- FILE \* **pLog**
- int \* **aiIndent**
- int **nIndent**
- int **iIndent**

The documentation for this struct was generated from the following file:

- shell.c

## 3.96 SortCtx Struct Reference

Collaboration diagram for SortCtx:

### Public Attributes

- [ExprList](#) \* **pOrderBy**
- int **nOBSat**
- int **iECursor**
- int **regReturn**
- int **labelBkOut**
- int **addrSortIndex**
- int **labelDone**
- u8 **sortFlags**
- u8 **bOrderedInnerLoop**

The documentation for this struct was generated from the following file:

- sqlite3.c

## 3.97 SorterFile Struct Reference

Collaboration diagram for SorterFile:

### Public Attributes

- [sqlite3\\_file](#) \* **pFd**
- i64 **iEof**

The documentation for this struct was generated from the following file:

- sqlite3.c

### 3.98 SorterList Struct Reference

Collaboration diagram for SorterList:

#### Public Attributes

- [SorterRecord](#) \* **pList**
- u8 \* **aMemory**
- int **szPMA**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

### 3.99 SorterRecord Struct Reference

Collaboration diagram for SorterRecord:

#### Public Attributes

- int **nVal**
- union {
  - [SorterRecord](#) \* **pNext**
  - int **iNext**
 } **u**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

### 3.100 SortSubtask Struct Reference

Collaboration diagram for SortSubtask:

#### Public Attributes

- [SQLiteThread](#) \* **pThread**
- int **bDone**
- [VdbeSorter](#) \* **pSorter**
- [UnpackedRecord](#) \* **pUnpacked**
- [SorterList](#) **list**
- int **nPMA**
- SorterCompare **xCompare**
- [SorterFile](#) **file**
- [SorterFile](#) **file2**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)



## 3.101 sqlite3 Struct Reference

Collaboration diagram for sqlite3:

### Classes

- struct [sqlite3InitInfo](#)

### Public Attributes

- [sqlite3\\_vfs](#) \* **pVfs**
- struct [Vdbe](#) \* **pVdbe**
- [CollSeq](#) \* **pDfltColl**
- [sqlite3\\_mutex](#) \* **mutex**
- [Db](#) \* **aDb**
- int **nDb**
- int **flags**
- i64 **lastRowid**
- i64 **szMmap**
- unsigned int **openFlags**
- int **errCode**
- int **errMask**
- int **iSysErrno**
- u16 **dbOptFlags**
- u8 **enc**
- u8 **autoCommit**
- u8 **temp\_store**
- u8 **mallocFailed**
- u8 **bBenignMalloc**
- u8 **dfltLockMode**
- signed char **nextAutovac**
- u8 **suppressErr**
- u8 **vtabOnConflict**
- u8 **isTransactionSavepoint**
- u8 **mTrace**
- int **nextPagesize**
- u32 **magic**
- int **nChange**
- int **nTotalChange**
- int **aLimit** [SQLITE\_N\_LIMIT]
- int **nMaxSorterMmap**
- struct [sqlite3::sqlite3InitInfo](#) **init**
- int **nVdbeActive**
- int **nVdbeRead**
- int **nVdbeWrite**
- int **nVdbeExec**
- int **nVDestroy**
- int **nExtension**
- void \*\* **aExtension**
- int(\* **xTrace**)(u32, void \*, void \*, void \*)
- void \* **pTraceArg**
- void(\* **xProfile**)(void \*, const char \*, u64)

- void \* **pProfileArg**
- void \* **pCommitArg**
- int(\* **xCommitCallback** )(void \*)
- void \* **pRollbackArg**
- void(\* **xRollbackCallback** )(void \*)
- void \* **pUpdateArg**
- void(\* **xUpdateCallback** )(void \*, int, const char \*, const char \*, sqlite\_int64)
- int(\* **xWalCallback** )(void \*, [sqlite3](#) \*, const char \*, int)
- void \* **pWalArg**
- void(\* **xCollNeeded** )(void \*, [sqlite3](#) \*, int eTextRep, const char \*)
- void(\* **xCollNeeded16** )(void \*, [sqlite3](#) \*, int eTextRep, const void \*)
- void \* **pCollNeededArg**
- [sqlite3\\_value](#) \* **pErr**
- union {
  - volatile int **isInterrupted**
  - double **notUsed1**
- } **u1**
- [Lookaside](#) **lookaside**
- [sqlite3\\_xauth](#) **xAuth**
- void \* **pAuthArg**
- int(\* **xProgress** )(void \*)
- void \* **pProgressArg**
- unsigned **nProgressOps**
- int **nVTrans**
- [Hash](#) **aModule**
- [VtabCtx](#) \* **pVtabCtx**
- [VTable](#) \*\* **aVTrans**
- [VTable](#) \* **pDisconnect**
- [Hash](#) **aFunc**
- [Hash](#) **aCollSeq**
- [BusyHandler](#) **busyHandler**
- [Db](#) **aDbStatic** [2]
- [Savepoint](#) \* **pSavepoint**
- int **busyTimeout**
- int **nSavepoint**
- int **nStatement**
- i64 **nDeferredCons**
- i64 **nDeferredImmCons**
- int \* **pnBytesFreed**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

### 3.102 [sqlite3\\_api\\_routines](#) Struct Reference

Collaboration diagram for [sqlite3\\_api\\_routines](#):

## Public Attributes

- void **(\* aggregate\_context )**(sqlite3\_context \*, int nBytes)
- int **(\* aggregate\_count )**(sqlite3\_context \*)
- int **(\* bind\_blob )**(sqlite3\_stmt \*, int, const void \*, int n, void(\*) (void \*))
- int **(\* bind\_double )**(sqlite3\_stmt \*, int, double)
- int **(\* bind\_int )**(sqlite3\_stmt \*, int, int)
- int **(\* bind\_int64 )**(sqlite3\_stmt \*, int, sqlite\_int64)
- int **(\* bind\_null )**(sqlite3\_stmt \*, int)
- int **(\* bind\_parameter\_count )**(sqlite3\_stmt \*)
- int **(\* bind\_parameter\_index )**(sqlite3\_stmt \*, const char \*zName)
- const char **(\* bind\_parameter\_name )**(sqlite3\_stmt \*, int)
- int **(\* bind\_text )**(sqlite3\_stmt \*, int, const char \*, int n, void(\*) (void \*))
- int **(\* bind\_text16 )**(sqlite3\_stmt \*, int, const void \*, int, void(\*) (void \*))
- int **(\* bind\_value )**(sqlite3\_stmt \*, int, const sqlite3\_value \*)
- int **(\* busy\_handler )**(sqlite3 \*, int(\*) (void \*, int), void \*)
- int **(\* busy\_timeout )**(sqlite3 \*, int ms)
- int **(\* changes )**(sqlite3 \*)
- int **(\* close )**(sqlite3 \*)
- int **(\* collation\_needed )**(sqlite3 \*, void \*, void(\*) (void \*, sqlite3 \*, int eTextRep, const char \*))
- int **(\* collation\_needed16 )**(sqlite3 \*, void \*, void(\*) (void \*, sqlite3 \*, int eTextRep, const void \*))
- const void **(\* column\_blob )**(sqlite3\_stmt \*, int iCol)
- int **(\* column\_bytes )**(sqlite3\_stmt \*, int iCol)
- int **(\* column\_bytes16 )**(sqlite3\_stmt \*, int iCol)
- int **(\* column\_count )**(sqlite3\_stmt \*pStmt)
- const char **(\* column\_database\_name )**(sqlite3\_stmt \*, int)
- const void **(\* column\_database\_name16 )**(sqlite3\_stmt \*, int)
- const char **(\* column\_decltype )**(sqlite3\_stmt \*, int i)
- const void **(\* column\_decltype16 )**(sqlite3\_stmt \*, int)
- double **(\* column\_double )**(sqlite3\_stmt \*, int iCol)
- int **(\* column\_int )**(sqlite3\_stmt \*, int iCol)
- sqlite\_int64 **(\* column\_int64 )**(sqlite3\_stmt \*, int iCol)
- const char **(\* column\_name )**(sqlite3\_stmt \*, int)
- const void **(\* column\_name16 )**(sqlite3\_stmt \*, int)
- const char **(\* column\_origin\_name )**(sqlite3\_stmt \*, int)
- const void **(\* column\_origin\_name16 )**(sqlite3\_stmt \*, int)
- const char **(\* column\_table\_name )**(sqlite3\_stmt \*, int)
- const void **(\* column\_table\_name16 )**(sqlite3\_stmt \*, int)
- const unsigned char **(\* column\_text )**(sqlite3\_stmt \*, int iCol)
- const void **(\* column\_text16 )**(sqlite3\_stmt \*, int iCol)
- int **(\* column\_type )**(sqlite3\_stmt \*, int iCol)
- sqlite3\_value **(\* column\_value )**(sqlite3\_stmt \*, int iCol)
- void **(\* commit\_hook )**(sqlite3 \*, int(\*) (void \*), void \*)
- int **(\* complete )**(const char \*sql)
- int **(\* complete16 )**(const void \*sql)
- int **(\* create\_collation )**(sqlite3 \*, const char \*, int, void \*, int(\*) (void \*, int, const void \*, int, const void \*))
- int **(\* create\_collation16 )**(sqlite3 \*, const void \*, int, void \*, int(\*) (void \*, int, const void \*, int, const void \*))
- int **(\* create\_function )**(sqlite3 \*, const char \*, int, int, void \*, void(\*) (sqlite3\_context \*, int, sqlite3\_value \*\*), void(\*) (sqlite3\_context \*, int, sqlite3\_value \*\*), void(\*) (sqlite3\_context \*))
- int **(\* create\_function16 )**(sqlite3 \*, const void \*, int, int, void \*, void(\*) (sqlite3\_context \*, int, sqlite3\_value \*\*), void(\*) (sqlite3\_context \*, int, sqlite3\_value \*\*), void(\*) (sqlite3\_context \*))
- int **(\* create\_module )**(sqlite3 \*, const char \*, const sqlite3\_module \*, void \*)
- int **(\* data\_count )**(sqlite3\_stmt \*pStmt)
- sqlite3 **(\* db\_handle )**(sqlite3\_stmt \*)
- int **(\* declare\_vtab )**(sqlite3 \*, const char \*)

- `int(* enable_shared_cache )(int)`
- `int(* errcode )(sqlite3 *db)`
- `const char *(* errmsg )(sqlite3 *)`
- `const void *(* errmsg16 )(sqlite3 *)`
- `int(* exec )(sqlite3 *, const char *, sqlite3_callback, void *, char **)`
- `int(* expired )(sqlite3_stmt *)`
- `int(* finalize )(sqlite3_stmt *pStmt)`
- `void(* free )(void *)`
- `void(* free_table )(char **result)`
- `int(* get_autocommit )(sqlite3 *)`
- `void(* get_auxdata )(sqlite3_context *, int)`
- `int(* get_table )(sqlite3 *, const char *, char ***, int *, int *, char **)`
- `int(* global_recover )(void)`
- `void(* interruptx )(sqlite3 *)`
- `sqlite_int64(* last_insert_rowid )(sqlite3 *)`
- `const char *(* libversion )(void)`
- `int(* libversion_number )(void)`
- `void(* malloc )(int)`
- `char *(* mprintf )(const char *,...)`
- `int(* open )(const char *, sqlite3 **)`
- `int(* open16 )(const void *, sqlite3 **)`
- `int(* prepare )(sqlite3 *, const char *, int, sqlite3_stmt **, const char **)`
- `int(* prepare16 )(sqlite3 *, const void *, int, sqlite3_stmt **, const void **)`
- `void(* profile )(sqlite3 *, void*)(void *, const char *, sqlite_uint64, void *)`
- `void(* progress_handler )(sqlite3 *, int, int*)(void *, void *)`
- `void(* realloc )(void *, int)`
- `int(* reset )(sqlite3_stmt *pStmt)`
- `void(* result_blob )(sqlite3_context *, const void *, int, void*)(void *)`
- `void(* result_double )(sqlite3_context *, double)`
- `void(* result_error )(sqlite3_context *, const char *, int)`
- `void(* result_error16 )(sqlite3_context *, const void *, int)`
- `void(* result_int )(sqlite3_context *, int)`
- `void(* result_int64 )(sqlite3_context *, sqlite_int64)`
- `void(* result_null )(sqlite3_context *)`
- `void(* result_text )(sqlite3_context *, const char *, int, void*)(void *)`
- `void(* result_text16 )(sqlite3_context *, const void *, int, void*)(void *)`
- `void(* result_text16be )(sqlite3_context *, const void *, int, void*)(void *)`
- `void(* result_text16le )(sqlite3_context *, const void *, int, void*)(void *)`
- `void(* result_value )(sqlite3_context *, sqlite3_value *)`
- `void(* rollback_hook )(sqlite3 *, void*)(void *, void *)`
- `int(* set_authorizer )(sqlite3 *, int*)(void *, int, const char *, const char *, const char *, const char *), void *`
- `void(* set_auxdata )(sqlite3_context *, int, void *, void*)(void *)`
- `char *(* snprintf )(int, char *, const char *,...)`
- `int(* step )(sqlite3_stmt *)`
- `int(* table_column_metadata )(sqlite3 *, const char *, const char *, const char *, char const **, char const **, int *, int *, int *)`
- `void(* thread_cleanup )(void)`
- `int(* total_changes )(sqlite3 *)`
- `void(* trace )(sqlite3 *, void(*xTrace)(void *, const char *), void *)`
- `int(* transfer_bindings )(sqlite3_stmt *, sqlite3_stmt *)`
- `void(* update_hook )(sqlite3 *, void*)(void *, int, char const *, char const *, sqlite_int64), void *)`
- `void(* user_data )(sqlite3_context *)`
- `const void *(* value_blob )(sqlite3_value *)`
- `int(* value_bytes )(sqlite3_value *)`

- `int(* value_bytes16 )(sqlite3_value *)`
- `double(* value_double )(sqlite3_value *)`
- `int(* value_int )(sqlite3_value *)`
- `sqlite_int64(* value_int64 )(sqlite3_value *)`
- `int(* value_numeric_type )(sqlite3_value *)`
- `const unsigned char *(* value_text )(sqlite3_value *)`
- `const void *(* value_text16 )(sqlite3_value *)`
- `const void *(* value_text16be )(sqlite3_value *)`
- `const void *(* value_text16le )(sqlite3_value *)`
- `int(* value_type )(sqlite3_value *)`
- `char *(* vprintf )(const char *, va_list)`
- `int(* overload_function )(sqlite3 *, const char *zFuncName, int nArg)`
- `int(* prepare_v2 )(sqlite3 *, const char *, int, sqlite3_stmt **, const char **)`
- `int(* prepare16_v2 )(sqlite3 *, const void *, int, sqlite3_stmt **, const void **)`
- `int(* clear_bindings )(sqlite3_stmt *)`
- `int(* create_module_v2 )(sqlite3 *, const char *, const sqlite3_module *, void *, void(*xDestroy)(void *))`
- `int(* bind_zeroblob )(sqlite3_stmt *, int, int)`
- `int(* blob_bytes )(sqlite3_blob *)`
- `int(* blob_close )(sqlite3_blob *)`
- `int(* blob_open )(sqlite3 *, const char *, const char *, const char *, sqlite3_int64, int, sqlite3_blob **)`
- `int(* blob_read )(sqlite3_blob *, void *, int, int)`
- `int(* blob_write )(sqlite3_blob *, const void *, int, int)`
- `int(* create_collation_v2 )(sqlite3 *, const char *, int, void *, int*)(void *, int, const void *, int, const void *, void*)(void *)`
- `int(* file_control )(sqlite3 *, const char *, int, void *)`
- `sqlite3_int64(* memory_highwater )(int)`
- `sqlite3_int64(* memory_used )(void)`
- `sqlite3_mutex *(* mutex_alloc )(int)`
- `void(* mutex_enter )(sqlite3_mutex *)`
- `void(* mutex_free )(sqlite3_mutex *)`
- `void(* mutex_leave )(sqlite3_mutex *)`
- `int(* mutex_try )(sqlite3_mutex *)`
- `int(* open_v2 )(const char *, sqlite3 **, int, const char *)`
- `int(* release_memory )(int)`
- `void(* result_error_nomem )(sqlite3_context *)`
- `void(* result_error_toobig )(sqlite3_context *)`
- `int(* sleep )(int)`
- `void(* soft_heap_limit )(int)`
- `sqlite3_vfs *(* vfs_find )(const char *)`
- `int(* vfs_register )(sqlite3_vfs *, int)`
- `int(* vfs_unregister )(sqlite3_vfs *)`
- `int(* xthreadsafe )(void)`
- `void(* result_zeroblob )(sqlite3_context *, int)`
- `void(* result_error_code )(sqlite3_context *, int)`
- `int(* test_control )(int,...)`
- `void(* randomness )(int, void *)`
- `sqlite3 *(* context_db_handle )(sqlite3_context *)`
- `int(* extended_result_codes )(sqlite3 *, int)`
- `int(* limit )(sqlite3 *, int, int)`
- `sqlite3_stmt *(* next_stmt )(sqlite3 *, sqlite3_stmt *)`
- `const char *(* sql )(sqlite3_stmt *)`
- `int(* status )(int, int *, int *, int)`
- `int(* backup_finish )(sqlite3_backup *)`
- `sqlite3_backup *(* backup_init )(sqlite3 *, const char *, sqlite3 *, const char *)`
- `int(* backup_pagecount )(sqlite3_backup *)`

- `int(* backup_remaining )(sqlite3_backup *)`
- `int(* backup_step )(sqlite3_backup *, int)`
- `const char *(* compileoption_get )(int)`
- `int(* compileoption_used )(const char *)`
- `int(* create_function_v2 )(sqlite3 *, const char *, int, int, void *, void(*xFunc)(sqlite3_context *, int, sqlite3_value **), void(*xStep)(sqlite3_context *, int, sqlite3_value **), void(*xFinal)(sqlite3_context *), void(*xDestroy)(void *))`
- `int(* db_config )(sqlite3 *, int,...)`
- `sqlite3_mutex *(* db_mutex )(sqlite3 *)`
- `int(* db_status )(sqlite3 *, int, int *, int *, int)`
- `int(* extended_errcode )(sqlite3 *)`
- `void(* log )(int, const char *,...)`
- `sqlite3_int64(* soft_heap_limit64 )(sqlite3_int64)`
- `const char *(* sourceid )(void)`
- `int(* stmt_status )(sqlite3_stmt *, int, int)`
- `int(* strnicmp )(const char *, const char *, int)`
- `int(* unlock_notify )(sqlite3 *, void(*) (void **, int), void *)`
- `int(* wal_autocheckpoint )(sqlite3 *, int)`
- `int(* wal_checkpoint )(sqlite3 *, const char *)`
- `void *(* wal_hook )(sqlite3 *, int(*) (void *, sqlite3 *, const char *, int), void *)`
- `int(* blob_reopen )(sqlite3_blob *, sqlite3_int64)`
- `int(* vtab_config )(sqlite3 *, int op,...)`
- `int(* vtab_on_conflict )(sqlite3 *)`
- `int(* close_v2 )(sqlite3 *)`
- `const char *(* db_filename )(sqlite3 *, const char *)`
- `int(* db_readonly )(sqlite3 *, const char *)`
- `int(* db_release_memory )(sqlite3 *)`
- `const char *(* errstr )(int)`
- `int(* stmt_busy )(sqlite3_stmt *)`
- `int(* stmt_readonly )(sqlite3_stmt *)`
- `int(* strcmp )(const char *, const char *)`
- `int(* uri_boolean )(const char *, const char *, int)`
- `sqlite3_int64(* uri_int64 )(const char *, const char *, sqlite3_int64)`
- `const char *(* uri_parameter )(const char *, const char *)`
- `char *(* vsnprintf )(int, char *, const char *, va_list)`
- `int(* wal_checkpoint_v2 )(sqlite3 *, const char *, int, int *, int *)`
- `int(* auto_extension )(void(*) (void))`
- `int(* bind_blob64 )(sqlite3_stmt *, int, const void *, sqlite3_uint64, void(*) (void *))`
- `int(* bind_text64 )(sqlite3_stmt *, int, const char *, sqlite3_uint64, void(*) (void *), unsigned char)`
- `int(* cancel_auto_extension )(void(*) (void))`
- `int(* load_extension )(sqlite3 *, const char *, const char *, char **)`
- `void *(* malloc64 )(sqlite3_uint64)`
- `sqlite3_uint64(* msize )(void *)`
- `void *(* realloc64 )(void *, sqlite3_uint64)`
- `void(* reset_auto_extension )(void)`
- `void(* result_blob64 )(sqlite3_context *, const void *, sqlite3_uint64, void(*) (void *))`
- `void(* result_text64 )(sqlite3_context *, const char *, sqlite3_uint64, void(*) (void *), unsigned char)`
- `int(* strcmp )(const char *, const char *)`
- `sqlite3_value *(* value_dup )(const sqlite3_value *)`
- `void(* value_free )(sqlite3_value *)`
- `int(* result_zeroblob64 )(sqlite3_context *, sqlite3_uint64)`
- `int(* bind_zeroblob64 )(sqlite3_stmt *, int, sqlite3_uint64)`
- `unsigned int(* value_subtype )(sqlite3_value *)`
- `void(* result_subtype )(sqlite3_context *, unsigned int)`
- `int(* status64 )(int, sqlite3_int64 *, sqlite3_int64 *, int)`

- `int(* strlike )(const char *, const char *, unsigned int)`
- `int(* db_cacheflush )(sqlite3 *)`
- `int(* system_errno )(sqlite3 *)`
- `int(* trace_v2 )(sqlite3 *, unsigned, int*)(unsigned, void *, void *, void *), void *)`
- `char *(* expanded_sql )(sqlite3_stmt *)`

The documentation for this struct was generated from the following files:

- `sqlite3.c`
- `sqlite3ext.h`

### 3.103 `sqlite3_backup` Struct Reference

Collaboration diagram for `sqlite3_backup`:

#### Public Attributes

- `sqlite3 * pDestDb`
- `Btree * pDest`
- `u32 iDestSchema`
- `int bDestLocked`
- `Pgno iNext`
- `sqlite3 * pSrcDb`
- `Btree * pSrc`
- `int rc`
- `Pgno nRemaining`
- `Pgno nPagecount`
- `int isAttached`
- `sqlite3_backup * pNext`

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.104 `sqlite3_context` Struct Reference

Collaboration diagram for `sqlite3_context`:

#### Public Attributes

- `Mem * pOut`
- `FuncDef * pFunc`
- `Mem * pMem`
- `Vdbe * pVdbe`
- `int iOp`
- `int isError`
- `u8 skipFlag`
- `u8 fErrorOrAux`
- `u8 argc`
- `sqlite3_value * argv [1]`

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.105 `sqlite3_file` Struct Reference

Collaboration diagram for `sqlite3_file`:

#### Public Attributes

- const struct `sqlite3_io_methods` \* **pMethods**

The documentation for this struct was generated from the following files:

- `sqlite3.c`
- `sqlite3.h`

### 3.106 `sqlite3_index_info::sqlite3_index_constraint` Struct Reference

#### Public Attributes

- int **iColumn**
- unsigned char **op**
- unsigned char **usable**
- int **iTermOffset**

The documentation for this struct was generated from the following files:

- `sqlite3.c`
- `sqlite3.h`

### 3.107 `sqlite3_index_info::sqlite3_index_constraint_usage` Struct Reference

#### Public Attributes

- int **argvIndex**
- unsigned char **omit**

The documentation for this struct was generated from the following files:

- `sqlite3.c`
- `sqlite3.h`

### 3.108 `sqlite3_index_info` Struct Reference

Collaboration diagram for `sqlite3_index_info`:



## Classes

- struct [sqlite3\\_index\\_constraint](#)
- struct [sqlite3\\_index\\_constraint\\_usage](#)
- struct [sqlite3\\_index\\_orderby](#)

## Public Attributes

- int **nConstraint**
- struct [sqlite3\\_index\\_info::sqlite3\\_index\\_constraint](#) \* **aConstraint**
- int **nOrderBy**
- struct [sqlite3\\_index\\_info::sqlite3\\_index\\_orderby](#) \* **aOrderBy**
- struct [sqlite3\\_index\\_info::sqlite3\\_index\\_constraint\\_usage](#) \* **aConstraintUsage**
- int **idxNum**
- char \* **idxStr**
- int **needToFreeIdxStr**
- int **orderByConsumed**
- double **estimatedCost**
- sqlite3\_int64 **estimatedRows**
- int **idxFlags**
- sqlite3\_uint64 **colUsed**

The documentation for this struct was generated from the following files:

- [sqlite3.c](#)
- [sqlite3.h](#)

## 3.109 sqlite3\_index\_info::sqlite3\_index\_orderby Struct Reference

### Public Attributes

- int **iColumn**
- unsigned char **desc**

The documentation for this struct was generated from the following files:

- [sqlite3.c](#)
- [sqlite3.h](#)

### 3.110 `sqlite3_io_methods` Struct Reference

#### Public Attributes

- `int iVersion`
- `int(* xClose )(sqlite3_file *)`
- `int(* xRead )(sqlite3_file *, void *, int iAmt, sqlite3_int64 iOfst)`
- `int(* xWrite )(sqlite3_file *, const void *, int iAmt, sqlite3_int64 iOfst)`
- `int(* xTruncate )(sqlite3_file *, sqlite3_int64 size)`
- `int(* xSync )(sqlite3_file *, int flags)`
- `int(* xFileSize )(sqlite3_file *, sqlite3_int64 *pSize)`
- `int(* xLock )(sqlite3_file *, int)`
- `int(* xUnlock )(sqlite3_file *, int)`
- `int(* xCheckReservedLock )(sqlite3_file *, int *pResOut)`
- `int(* xFileControl )(sqlite3_file *, int op, void *pArg)`
- `int(* xSectorSize )(sqlite3_file *)`
- `int(* xDeviceCharacteristics )(sqlite3_file *)`
- `int(* xShmMap )(sqlite3_file *, int iPg, int pgsz, int, void volatile **)`
- `int(* xShmLock )(sqlite3_file *, int offset, int n, int flags)`
- `void(* xShmBarrier )(sqlite3_file *)`
- `int(* xShmUnmap )(sqlite3_file *, int deleteFlag)`
- `int(* xFetch )(sqlite3_file *, sqlite3_int64 iOfst, int iAmt, void **pp)`
- `int(* xUnfetch )(sqlite3_file *, sqlite3_int64 iOfst, void *p)`

The documentation for this struct was generated from the following files:

- `sqlite3.c`
- `sqlite3.h`

### 3.111 `sqlite3_mem_methods` Struct Reference

#### Public Attributes

- `void *(* xMalloc )(int)`
- `void(* xFree )(void *)`
- `void *(* xRealloc )(void *, int)`
- `int(* xSize )(void *)`
- `int(* xRoundup )(int)`
- `int(* xInit )(void *)`
- `void(* xShutdown )(void *)`
- `void * pAppData`

The documentation for this struct was generated from the following files:

- `sqlite3.c`
- `sqlite3.h`

## 3.112 sqlite3\_module Struct Reference

### Public Attributes

- int **iVersion**
- int(\* **xCreate**)(sqlite3 \*, void \*pAux, int argc, const char \*const \*argv, sqlite3\_vtab \*\*ppVTab, char \*\*)
- int(\* **xConnect**)(sqlite3 \*, void \*pAux, int argc, const char \*const \*argv, sqlite3\_vtab \*\*ppVTab, char \*\*)
- int(\* **xBestIndex**)(sqlite3\_vtab \*pVTab, sqlite3\_index\_info \*)
- int(\* **xDisconnect**)(sqlite3\_vtab \*pVTab)
- int(\* **xDestroy**)(sqlite3\_vtab \*pVTab)
- int(\* **xOpen**)(sqlite3\_vtab \*pVTab, sqlite3\_vtab\_cursor \*\*ppCursor)
- int(\* **xClose**)(sqlite3\_vtab\_cursor \*)
- int(\* **xFilter**)(sqlite3\_vtab\_cursor \*, int idxNum, const char \*idxStr, int argc, sqlite3\_value \*\*argv)
- int(\* **xNext**)(sqlite3\_vtab\_cursor \*)
- int(\* **xEOF**)(sqlite3\_vtab\_cursor \*)
- int(\* **xColumn**)(sqlite3\_vtab\_cursor \*, sqlite3\_context \*, int)
- int(\* **xRowid**)(sqlite3\_vtab\_cursor \*, sqlite3\_int64 \*pRowid)
- int(\* **xUpdate**)(sqlite3\_vtab \*, int, sqlite3\_value \*\*, sqlite3\_int64 \*)
- int(\* **xBegin**)(sqlite3\_vtab \*pVTab)
- int(\* **xSync**)(sqlite3\_vtab \*pVTab)
- int(\* **xCommit**)(sqlite3\_vtab \*pVTab)
- int(\* **xRollback**)(sqlite3\_vtab \*pVTab)
- int(\* **xFindFunction**)(sqlite3\_vtab \*pVTab, int nArg, const char \*zName, void(\*\*pFunc)(sqlite3\_context \*, int, sqlite3\_value \*\*), void \*\*ppArg)
- int(\* **xRename**)(sqlite3\_vtab \*pVTab, const char \*zNew)
- int(\* **xSavepoint**)(sqlite3\_vtab \*pVTab, int)
- int(\* **xRelease**)(sqlite3\_vtab \*pVTab, int)
- int(\* **xRollbackTo**)(sqlite3\_vtab \*pVTab, int)

The documentation for this struct was generated from the following files:

- sqlite3.c
- sqlite3.h

## 3.113 sqlite3\_mutex Struct Reference

### Public Attributes

- pthread\_mutex\_t **mutex**

The documentation for this struct was generated from the following file:

- sqlite3.c

## 3.114 sqlite3\_mutex\_methods Struct Reference

Collaboration diagram for sqlite3\_mutex\_methods:

## Public Attributes

- int(\* **xMutexInit** )(void)
- int(\* **xMutexEnd** )(void)
- [sqlite3\\_mutex](#) \*(**xMutexAlloc** )(int)
- void(\* **xMutexFree** )(sqlite3\_mutex \*)
- void(\* **xMutexEnter** )(sqlite3\_mutex \*)
- int(\* **xMutexTry** )(sqlite3\_mutex \*)
- void(\* **xMutexLeave** )(sqlite3\_mutex \*)
- int(\* **xMutexHeld** )(sqlite3\_mutex \*)
- int(\* **xMutexNotheld** )(sqlite3\_mutex \*)

The documentation for this struct was generated from the following files:

- [sqlite3.c](#)
- [sqlite3.h](#)

## 3.115 [sqlite3\\_pcache\\_methods](#) Struct Reference

### Public Attributes

- void \* **pArg**
- int(\* **xInit** )(void \*)
- void(\* **xShutdown** )(void \*)
- [sqlite3\\_pcache](#) \*(**xCreate** )(int szPage, int bPurgeable)
- void(\* **xCachesize** )(sqlite3\_pcache \*, int nCachesize)
- int(\* **xPagecount** )(sqlite3\_pcache \*)
- void(\* **xFetch** )(sqlite3\_pcache \*, unsigned key, int createFlag)
- void(\* **xUnpin** )(sqlite3\_pcache \*, void \*, int discard)
- void(\* **xRekey** )(sqlite3\_pcache \*, void \*, unsigned oldKey, unsigned newKey)
- void(\* **xTruncate** )(sqlite3\_pcache \*, unsigned iLimit)
- void(\* **xDestroy** )(sqlite3\_pcache \*)

The documentation for this struct was generated from the following files:

- [sqlite3.c](#)
- [sqlite3.h](#)

## 3.116 [sqlite3\\_pcache\\_methods2](#) Struct Reference

Collaboration diagram for [sqlite3\\_pcache\\_methods2](#):

### Public Attributes

- `int iVersion`
- `void * pArg`
- `int(* xInit )(void *)`
- `void(* xShutdown )(void *)`
- `sqlite3_pcache *(* xCreate )(int szPage, int szExtra, int bPurgeable)`
- `void(* xCachesize )(sqlite3_pcache *, int nCachesize)`
- `int(* xPagecount )(sqlite3_pcache *)`
- `sqlite3_pcache_page *(* xFetch )(sqlite3_pcache *, unsigned key, int createFlag)`
- `void(* xUnpin )(sqlite3_pcache *, sqlite3\_pcache\_page *, int discard)`
- `void(* xRekey )(sqlite3_pcache *, sqlite3\_pcache\_page *, unsigned oldKey, unsigned newKey)`
- `void(* xTruncate )(sqlite3_pcache *, unsigned iLimit)`
- `void(* xDestroy )(sqlite3_pcache *)`
- `void(* xShrink )(sqlite3_pcache *)`

The documentation for this struct was generated from the following files:

- `sqlite3.c`
- `sqlite3.h`

## 3.117 `sqlite3_pcache_page` Struct Reference

### Public Attributes

- `void * pBuf`
- `void * pExtra`

The documentation for this struct was generated from the following files:

- `sqlite3.c`
- `sqlite3.h`

## 3.118 `sqlite3_rtree_geometry` Struct Reference

### Public Attributes

- `void * pContext`
- `int nParam`
- `sqlite3_rtree_dbl * aParam`
- `void * pUser`
- `void(* xDelUser )(void *)`

The documentation for this struct was generated from the following files:

- `sqlite3.c`
- `sqlite3.h`

### 3.119 sqlite3\_rtree\_query\_info Struct Reference

Collaboration diagram for sqlite3\_rtree\_query\_info:

#### Public Attributes

- void \* **pContext**
- int **nParam**
- sqlite3\_rtree\_dbl \* **aParam**
- void \* **pUser**
- void(\* **xDelUser** )(void \*)
- sqlite3\_rtree\_dbl \* **aCoord**
- unsigned int \* **anQueue**
- int **nCoord**
- int **iLevel**
- int **mxLevel**
- sqlite3\_int64 **iRowid**
- sqlite3\_rtree\_dbl **rParentScore**
- int **eParentWithin**
- int **eWithin**
- sqlite3\_rtree\_dbl **rScore**
- [sqlite3\\_value](#) \*\* **apSqlParam**

The documentation for this struct was generated from the following files:

- [sqlite3.c](#)
- [sqlite3.h](#)

### 3.120 sqlite3\_vfs Struct Reference

Collaboration diagram for sqlite3\_vfs:

#### Public Attributes

- int **iVersion**
- int **szOsFile**
- int **mxPathname**
- [sqlite3\\_vfs](#) \* **pNext**
- const char \* **zName**
- void \* **pAppData**
- int(\* **xOpen** )(sqlite3\_vfs \*, const char \*zName, [sqlite3\\_file](#) \*, int flags, int \*pOutFlags)
- int(\* **xDelete** )(sqlite3\_vfs \*, const char \*zName, int syncDir)
- int(\* **xAccess** )(sqlite3\_vfs \*, const char \*zName, int flags, int \*pResOut)
- int(\* **xFullPathname** )(sqlite3\_vfs \*, const char \*zName, int nOut, char \*zOut)
- void(\* **xDIOpen** )(sqlite3\_vfs \*, const char \*zFilename)
- void(\* **xDLError** )(sqlite3\_vfs \*, int nByte, char \*zErrMsg)
- void(\* **xDISym** )(sqlite3\_vfs \*, void \*, const char \*zSymbol))(void)
- void(\* **xDIClose** )(sqlite3\_vfs \*, void \*)
- int(\* **xRandomness** )(sqlite3\_vfs \*, int nByte, char \*zOut)

- int(\* **xSleep** )(sqlite3\_vfs \*, int microseconds)
- int(\* **xCurrentTime** )(sqlite3\_vfs \*, double \*)
- int(\* **xGetLastError** )(sqlite3\_vfs \*, int, char \*)
- int(\* **xCurrentTimeInt64** )(sqlite3\_vfs \*, sqlite3\_int64 \*)
- int(\* **xSetSystemCall** )(sqlite3\_vfs \*, const char \*zName, sqlite3\_syscall\_ptr)
- sqlite3\_syscall\_ptr(\* **xGetSystemCall** )(sqlite3\_vfs \*, const char \*zName)
- const char \*(\* **xNextSystemCall** )(sqlite3\_vfs \*, const char \*zName)

The documentation for this struct was generated from the following files:

- sqlite3.c
- sqlite3.h

## 3.121 sqlite3\_vtab Struct Reference

Collaboration diagram for sqlite3\_vtab:

### Public Attributes

- const [sqlite3\\_module](#) \* **pModule**
- int **nRef**
- char \* **zErrMsg**

The documentation for this struct was generated from the following files:

- sqlite3.c
- sqlite3.h

## 3.122 sqlite3\_vtab\_cursor Struct Reference

Collaboration diagram for sqlite3\_vtab\_cursor:

### Public Attributes

- [sqlite3\\_vtab](#) \* **pVtab**

The documentation for this struct was generated from the following files:

- sqlite3.c
- sqlite3.h

## 3.123 Sqlite3Config Struct Reference

Collaboration diagram for Sqlite3Config:

## Public Attributes

- int **bMemstat**
- int **bCoreMutex**
- int **bFullMutex**
- int **bOpenUri**
- int **bUseCis**
- int **mxStrlen**
- int **neverCorrupt**
- int **szLookaside**
- int **nLookaside**
- int **nStmtSpill**
- [sqlite3\\_mem\\_methods](#) **m**
- [sqlite3\\_mutex\\_methods](#) **mutex**
- [sqlite3\\_pcache\\_methods2](#) **pcache2**
- void \* **pHeap**
- int **nHeap**
- int **mnReq**
- int **mxReq**
- [sqlite3\\_int64](#) **szMmap**
- [sqlite3\\_int64](#) **mxMmap**
- void \* **pScratch**
- int **szScratch**
- int **nScratch**
- void \* **pPage**
- int **szPage**
- int **nPage**
- int **mxParserStack**
- int **sharedCacheEnabled**
- u32 **szPma**
- int **isInit**
- int **inProgress**
- int **isMutexInit**
- int **isMallocInit**
- int **isPCacheInit**
- int **nReflnitMutex**
- [sqlite3\\_mutex](#) \* **plnitMutex**
- void(\* **xLog** )(void \*, int, const char \*)
- void \* **pLogArg**
- int(\* **xTestCallback** )(int)
- int **bLocaltimeFault**
- int **iOnceResetThreshold**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)



## 3.124 `sqlite3::sqlite3InitInfo` Struct Reference

### Public Attributes

- `int newTnum`
- `u8 iDb`
- `u8 busy`
- `u8 orphanTrigger`
- `u8 imposterTable`

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.125 `SQLiteThread` Struct Reference

### Public Attributes

- `pthread_t tid`
- `int done`
- `void * pOut`
- `void *(* xTask )(void *)`
- `void * pln`

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.126 `SrcCount` Struct Reference

Collaboration diagram for `SrcCount`:

### Public Attributes

- `SrcList * pSrc`
- `int nThis`
- `int nOther`

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.127 SrcList Struct Reference

Collaboration diagram for SrcList:

#### Classes

- struct [SrcList\\_item](#)

#### Public Attributes

- int **nSrc**
- u32 **nAlloc**
- struct [SrcList::SrcList\\_item](#) **a** [1]

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.128 SrcList::SrcList\_item Struct Reference

Collaboration diagram for SrcList::SrcList\_item:

#### Public Attributes

- [Schema](#) \* **pSchema**
- char \* **zDatabase**
- char \* **zName**
- char \* **zAlias**
- [Table](#) \* **pTab**
- [Select](#) \* **pSelect**
- int **addrFillSub**
- int **regReturn**
- int **regResult**
- struct {
  - u8 **jointype**
  - unsigned **notIndexed**:1
  - unsigned **isIndexedBy**:1
  - unsigned **isTabFunc**:1
  - unsigned **isCorrelated**:1
  - unsigned **viaCoroutine**:1
  - unsigned **isRecursive**:1
- u8 **iSelectId**
- int **iCursor**
- [Expr](#) \* **pOn**
- [IdList](#) \* **pUsing**
- Bitmask **colUsed**

- union {  
    char \* **zIndexedBy**  
    [ExprList](#) \* **pFuncArg**  
} **u1**
- [Index](#) \* **pIBIndex**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.129 Stat4Accum Struct Reference

Collaboration diagram for Stat4Accum:

### Public Attributes

- tRowcnt **nRow**
- tRowcnt **nPSample**
- int **nCol**
- int **nKeyCol**
- int **mxSample**
- [Stat4Sample](#) **current**
- u32 **iPrn**
- [Stat4Sample](#) \* **aBest**
- int **iMin**
- int **nSample**
- int **iGet**
- [Stat4Sample](#) \* **a**
- [sqlite3](#) \* **db**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.130 Stat4Sample Struct Reference

### Public Attributes

- tRowcnt \* **anEq**
- tRowcnt \* **anDLt**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.131 StrAccum Struct Reference

Collaboration diagram for StrAccum:

#### Public Attributes

- [sqlite3](#) \* **db**
- char \* **zBase**
- char \* **zText**
- u32 **nChar**
- u32 **nAlloc**
- u32 **mxAlloc**
- u8 **accError**
- u8 **printfFlags**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

### 3.132 SubProgram Struct Reference

Collaboration diagram for SubProgram:

#### Public Attributes

- [VdbeOp](#) \* **aOp**
- int **nOp**
- int **nMem**
- int **nCsr**
- void \* **token**
- [SubProgram](#) \* **pNext**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

### 3.133 SumCtx Struct Reference

#### Public Attributes

- double **rSum**
- i64 **iSum**
- i64 **cnt**
- u8 **overflow**
- u8 **approx**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.134 Table Struct Reference

Collaboration diagram for Table:

### Public Attributes

- char \* **zName**
- Column \* **aCol**
- Index \* **pIndex**
- Select \* **pSelect**
- FKey \* **pFKey**
- char \* **zColAff**
- ExprList \* **pCheck**
- int **tnum**
- i16 **iPKey**
- i16 **nCol**
- u16 **nRef**
- LogEst **nRowLogEst**
- LogEst **szTabRow**
- u8 **tabFlags**
- u8 **keyConf**
- int **addColOffset**
- int **nModuleArg**
- char \*\* **azModuleArg**
- VTable \* **pVTable**
- Trigger \* **pTrigger**
- Schema \* **pSchema**
- Table \* **pNextZombie**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.135 TableLock Struct Reference

### Public Attributes

- int **iDb**
- int **iTab**
- u8 **isWriteLock**
- const char \* **zName**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.136 TabResult Struct Reference

#### Public Attributes

- char \*\* **azResult**
- char \* **zErrMsg**
- u32 **nAlloc**
- u32 **nRow**
- u32 **nColumn**
- u32 **nData**
- int **rc**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.137 Token Struct Reference

#### Public Attributes

- const char \* **z**
- unsigned int **n**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.138 TrigEvent Struct Reference

Collaboration diagram for TrigEvent:

#### Public Attributes

- int **a**
- [IdList](#) \* **b**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.139 Trigger Struct Reference

Collaboration diagram for Trigger:

**Public Attributes**

- char \* **zName**
- char \* **table**
- u8 **op**
- u8 **tr\_tm**
- [Expr](#) \* **pWhen**
- [IdList](#) \* **pColumns**
- [Schema](#) \* **pSchema**
- [Schema](#) \* **pTabSchema**
- [TriggerStep](#) \* **step\_list**
- [Trigger](#) \* **pNext**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.140 TriggerPrg Struct Reference

Collaboration diagram for TriggerPrg:

**Public Attributes**

- [Trigger](#) \* **pTrigger**
- [TriggerPrg](#) \* **pNext**
- [SubProgram](#) \* **pProgram**
- int **orconf**
- u32 **aColmask** [2]

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.141 TriggerStep Struct Reference

Collaboration diagram for TriggerStep:

**Public Attributes**

- u8 **op**
- u8 **orconf**
- [Trigger](#) \* **pTrig**
- [Select](#) \* **pSelect**
- char \* **zTarget**
- [Expr](#) \* **pWhere**
- [ExprList](#) \* **pExprList**
- [IdList](#) \* **pIdList**
- [TriggerStep](#) \* **pNext**
- [TriggerStep](#) \* **pLast**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.142 Truck Class Reference

Truck Class.

```
#include <Truck.h>
```

### Public Member Functions

- [Truck](#) (const double)  
*Truck constructor.*
- int [getTruckID](#) () const  
*Getter for Truck ID Number.*
- double [getWeight](#) () const  
*Getter for truck's weight limit.*
- vector< [Package](#) \* > [getPackages](#) () const  
*Getter for package vector.*
- vector< string \* > [getDirections](#) () const  
*Getter for directions vector.*
- void [setDirections](#) (vector< string \* >)  
*Setter for directions vector.*
- void [addPackage](#) ([Package](#) \*)  
*Adds a package to the truck.*
- void [deliverPackage](#) ([Package](#) \*pack)  
*Delivers a package.*
- string [toString](#) () const  
*Returns a string representation of the truck and its contents.*

### Static Public Attributes

- static int [truckCount](#) = 0  
*A count of the total number of trucks on the road.*

### Private Attributes

- const int [truckID](#)  
*This truck's identification number.*
- double [weight](#)  
*This truck's weight limit.*
- double [currentWeight](#) = 0  
*The current weight of the truck's load of packages.*
- vector< [Package](#) \* > [packages](#)  
*Vector storing the truck's current load of packages.*
- vector< string \* > [directions](#)  
*Vector of strings storing the directions for this truck.*



### 3.142.1 Detailed Description

Truck Class.

This [Truck](#) class defines a truck. It is responsible for holding a list of all packages in its delivery queue and a list of instructions for delivering those packages.

### 3.142.2 Constructor & Destructor Documentation

#### 3.142.2.1 Truck::Truck ( const double w )

Truck constructor.

Constructor.

Parameters

<i>weight</i>	Weight limit
<i>w</i>	Weight limit for this truck

### 3.142.3 Member Function Documentation

#### 3.142.3.1 void Truck::addPackage ( Package \* pack )

Adds a package to the truck.

Add a package to the truck.

Adds a package to the truck and adds its weight to the current load

Parameters

<i>pack</i>	A package object
-------------	------------------

Adds a package to the truck. If this would go over the weight limit of the truck, then it throws an error. It also adds the weight of the package to the truck if the package fits.

Parameters

<i>*pack</i>	Pointer to package
--------------	--------------------

#### 3.142.3.2 void Truck::deliverPackage ( Package \* pack )

Delivers a package.

Delivers package.

Removes a package from the truck, subtracts its weight, and sets the package's delivered flag.

sets delivered flag to true if this package is in the vector

#### 3.142.3.3 `vector< string * > Truck::getDirections ( ) const`

Getter for directions vector.

Get directions assigned to this truck.

##### Returns

the vector containing the truck's current list of directions  
`vector<string*>`

#### 3.142.3.4 `vector< Package * > Truck::getPackages ( ) const`

Getter for package vector.

Gets truck's currently loaded packages.

return `vector<Package*>`

#### 3.142.3.5 `int Truck::getTruckID ( ) const`

Getter for [Truck](#) ID Number.

Gets this truck's ID number.

##### Returns

int [Truck](#) ID number

return int

#### 3.142.3.6 `double Truck::getWeight ( ) const`

Getter for truck's weight limit.

Gets truck's weight limit.

##### Returns

double [Truck](#)'s weight limit  
double

#### 3.142.3.7 `void Truck::setDirections ( vector< string * > direct )`

Setter for directions vector.

## Parameters

<i>directions</i>	a pointer to a vector containing the truck's assigned directions
<i>direct</i>	pointer to directions vector

## 3.142.3.8 string Truck::toString ( ) const

Returns a string representation of the truck and its contents.

To string.

## Returns

string A string representation of the truck and its contents

## 3.142.4 Member Data Documentation

## 3.142.4.1 int Truck::truckCount = 0 [static]

A count of the total number of trucks on the road.

Initialize truck count.

The documentation for this class was generated from the following files:

- Truck.h
- Truck.cpp

## 3.143 unixFile Struct Reference

Collaboration diagram for unixFile:

## Public Attributes

- [sqlite3\\_io\\_methods](#) const \* **pMethod**
- [sqlite3\\_vfs](#) \* **pVfs**
- [unixInodeInfo](#) \* **plnode**
- int **h**
- unsigned char **eFileLock**
- unsigned short int **ctrlFlags**
- int **lastErrno**
- void \* **lockingContext**
- [UnixUnusedFd](#) \* **pUnused**
- const char \* **zPath**
- [unixShm](#) \* **pShm**
- int **szChunk**

The documentation for this struct was generated from the following file:

- sqlite3.c

### 3.144 unixFileId Struct Reference

#### Public Attributes

- dev\_t **dev**
- ino\_t **ino**

The documentation for this struct was generated from the following file:

- sqlite3.c

### 3.145 unixNodeInfo Struct Reference

Collaboration diagram for unixNodeInfo:

#### Public Attributes

- struct [unixFileId](#) **fileId**
- int **nShared**
- unsigned char **eFileLock**
- unsigned char **bProcessLock**
- int **nRef**
- [unixShmNode](#) \* **pShmNode**
- int **nLock**
- [UnixUnusedFd](#) \* **pUnused**
- [unixNodeInfo](#) \* **pNext**
- [unixNodeInfo](#) \* **pPrev**

The documentation for this struct was generated from the following file:

- sqlite3.c

### 3.146 unixShm Struct Reference

Collaboration diagram for unixShm:

#### Public Attributes

- [unixShmNode](#) \* **pShmNode**
- [unixShm](#) \* **pNext**
- u8 **hasMutex**
- u8 **id**
- u16 **sharedMask**
- u16 **exclMask**

The documentation for this struct was generated from the following file:

- sqlite3.c

## 3.147 unixShmNode Struct Reference

Collaboration diagram for unixShmNode:

### Public Attributes

- [unixInodeInfo](#) \* **pInode**
- [sqlite3\\_mutex](#) \* **mutex**
- char \* **zFilename**
- int **h**
- int **szRegion**
- u16 **nRegion**
- u8 **isReadonly**
- char \*\* **apRegion**
- int **nRef**
- [unixShm](#) \* **pFirst**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.148 UnixUnusedFd Struct Reference

Collaboration diagram for UnixUnusedFd:

### Public Attributes

- int **fd**
- int **flags**
- [UnixUnusedFd](#) \* **pNext**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.149 UnpackedRecord Struct Reference

Collaboration diagram for UnpackedRecord:

## Public Attributes

- [KeyInfo](#) \* **pKeyInfo**
- [Mem](#) \* **aMem**
- u16 **nField**
- i8 **default\_rc**
- u8 **errCode**
- i8 **r1**
- i8 **r2**
- u8 **eqSeen**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.150 ValueNewStat4Ctx Struct Reference

Collaboration diagram for ValueNewStat4Ctx:

## Public Attributes

- [Parse](#) \* **pParse**
- [Index](#) \* **pIdx**
- [UnpackedRecord](#) \*\* **ppRec**
- int **iVal**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.151 Vdbe Struct Reference

Collaboration diagram for Vdbe:

## Public Attributes

- [sqlite3](#) \* **db**
- [Vdbe](#) \* **pPrev**
- [Vdbe](#) \* **pNext**
- [Parse](#) \* **pParse**
- ynVar **nVar**
- ynVar **nzVar**
- u32 **magic**
- int **nMem**
- int **nCursor**
- u32 **cacheCtr**
- int **pc**
- int **rc**
- int **nChange**
- int **iStatement**
- i64 **iCurrentTime**
- i64 **nFkConstraint**
- i64 **nStmtDefCons**
- i64 **nStmtDeflmmCons**
- [Op](#) \* **aOp**
- [Mem](#) \* **aMem**
- [Mem](#) \*\* **apArg**
- [Mem](#) \* **aColName**
- [Mem](#) \* **pResultSet**
- char \* **zErrMsg**
- [VdbeCursor](#) \*\* **apCsr**
- [Mem](#) \* **aVar**
- char \*\* **azVar**
- i64 **startTime**
- int **nOp**
- u16 **nResColumn**
- u8 **errorAction**
- u8 **minWriteFileFormat**
- bft **expired**:1
- bft **doingRerun**:1
- bft **explain**:2
- bft **changeCntOn**:1
- bft **runOnlyOnce**:1
- bft **usesStmtJournal**:1
- bft **readOnly**:1
- bft **blsReader**:1
- bft **isPrepareV2**:1
- yDbMask **btreeMask**
- yDbMask **lockMask**
- u32 **aCounter** [5]
- char \* **zSql**
- void \* **pFree**
- [VdbeFrame](#) \* **pFrame**
- [VdbeFrame](#) \* **pDelFrame**
- int **nFrame**
- u32 **expmask**
- [SubProgram](#) \* **pProgram**
- [AuxData](#) \* **pAuxData**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

### 3.152 VdbeCursor Struct Reference

Collaboration diagram for VdbeCursor:

#### Public Attributes

- u8 **eCurType**
- i8 **iDb**
- u8 **nullRow**
- u8 **deferredMoveto**
- u8 **isTable**
- Bool **isEphemeral**:1
- Bool **useRandomRowid**:1
- Bool **isOrdered**:1
- Pgeno **pgnoRoot**
- i16 **nField**
- u16 **nHdrParsed**
- union {
  - [BtCursor](#) \* **pCursor**
  - [sqlite3\\_vtab\\_cursor](#) \* **pVCur**
  - int **pseudoTableReg**
  - [VdbeSorter](#) \* **pSorter**
- } **uc**
- [Btree](#) \* **pBt**
- [KeyInfo](#) \* **pKeyInfo**
- int **seekResult**
- i64 **seqCount**
- i64 **movetoTarget**
- [VdbeCursor](#) \* **pAltCursor**
- int \* **aAltMap**
- u32 **cacheStatus**
- u32 **payloadSize**
- u32 **szRow**
- u32 **iHdrOffset**
- const u8 \* **aRow**
- u32 \* **aOffset**
- u32 **aType** [1]

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

### 3.153 VdbeFrame Struct Reference

Collaboration diagram for VdbeFrame:



### Public Attributes

- [Vdbe](#) \* **v**
- [VdbeFrame](#) \* **pParent**
- [Op](#) \* **aOp**
- i64 \* **anExec**
- [Mem](#) \* **aMem**
- [VdbeCursor](#) \*\* **apCsr**
- void \* **token**
- i64 **lastRowid**
- [AuxData](#) \* **pAuxData**
- int **nCursor**
- int **pc**
- int **nOp**
- int **nMem**
- int **nChildMem**
- int **nChildCsr**
- int **nChange**
- int **nDbChange**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.154 VdbeOp Struct Reference

Collaboration diagram for VdbeOp:

### Classes

- union [p4union](#)

### Public Attributes

- u8 **opcode**
- signed char **p4type**
- u8 **notUsed1**
- u8 **p5**
- int **p1**
- int **p2**
- int **p3**
- union [VdbeOp::p4union](#) **p4**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

### 3.155 VdbeOpList Struct Reference

#### Public Attributes

- u8 **opcode**
- signed char **p1**
- signed char **p2**
- signed char **p3**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.156 VdbeSorter Struct Reference

Collaboration diagram for VdbeSorter:

#### Public Attributes

- int **mnPmaSize**
- int **mxPmaSize**
- int **mxKeysize**
- int **pgsz**
- [PmaReader](#) \* **pReader**
- [MergeEngine](#) \* **pMerger**
- [sqlite3](#) \* **db**
- [KeyInfo](#) \* **pKeyInfo**
- [UnpackedRecord](#) \* **pUnpacked**
- [SorterList](#) **list**
- int **iMemory**
- int **nMemory**
- u8 **bUsePMA**
- u8 **bUseThreads**
- u8 **iPrev**
- u8 **nTask**
- u8 **typeMask**
- [SortSubtask](#) **aTask** [1]

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.157 VtabCtx Struct Reference

Collaboration diagram for VtabCtx:

### Public Attributes

- [VTable](#) \* **pVTable**
- [Table](#) \* **pTab**
- [VtabCtx](#) \* **pPrior**
- int **bDeclared**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.158 VTable Struct Reference

Collaboration diagram for VTable:

### Public Attributes

- [sqlite3](#) \* **db**
- [Module](#) \* **pMod**
- [sqlite3\\_vtab](#) \* **pVtab**
- int **nRef**
- u8 **bConstraint**
- int **iSavepoint**
- [VTable](#) \* **pNext**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.159 vxworksFileId Struct Reference

Collaboration diagram for vxworksFileId:

### Public Attributes

- struct [vxworksFileId](#) \* **pNext**
- int **nRef**
- int **nName**
- char \* **zCanonicalName**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.160 Wal Struct Reference

Collaboration diagram for Wal:

#### Public Attributes

- [sqlite3\\_vfs](#) \* **pVfs**
- [sqlite3\\_file](#) \* **pDbFd**
- [sqlite3\\_file](#) \* **pWalFd**
- u32 **iCallback**
- i64 **mxWalSize**
- int **nWiData**
- int **szFirstBlock**
- volatile u32 \*\* **apWiData**
- u32 **szPage**
- i16 **readLock**
- u8 **syncFlags**
- u8 **exclusiveMode**
- u8 **writeLock**
- u8 **ckptLock**
- u8 **readOnly**
- u8 **truncateOnCommit**
- u8 **syncHeader**
- u8 **padToSectorBoundary**
- [WallIndexHdr](#) **hdr**
- u32 **minFrame**
- u32 **iReCksum**
- const char \* **zWalName**
- u32 **nCkpt**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

### 3.161 WalCkptInfo Struct Reference

#### Public Attributes

- u32 **nBackfill**
- u32 **aReadMark** [WAL\_NREADER]
- u8 **aLock** [SQLITE\_SHM\_NLOCK]
- u32 **nBackfillAttempted**
- u32 **notUsed0**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.162 WalIndexHdr Struct Reference

### Public Attributes

- u32 **iVersion**
- u32 **unused**
- u32 **iChange**
- u8 **isInit**
- u8 **bigEndCksum**
- u16 **szPage**
- u32 **mxFrame**
- u32 **nPage**
- u32 **aFrameCksum** [2]
- u32 **aSalt** [2]
- u32 **aCksum** [2]

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.163 WalIterator Struct Reference

Collaboration diagram for WalIterator:

### Classes

- struct [WalSegment](#)

### Public Attributes

- int **iPrior**
- int **nSegment**
- struct [WalIterator::WalSegment](#) **aSegment** [1]

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.164 Walker Struct Reference

Collaboration diagram for Walker:

### Public Attributes

- [Parse](#) \* **pParse**
- int(\* **xExprCallback** )(Walker \*, Expr \*)
- int(\* **xSelectCallback** )(Walker \*, Select \*)
- void(\* **xSelectCallback2** )(Walker \*, Select \*)
- int **walkerDepth**
- u8 **eCode**
- union {
  - [NameContext](#) \* **pNC**
  - int **n**
  - int **iCur**
  - [SrcList](#) \* **pSrcList**
  - struct [SrcCount](#) \* **pSrcCount**
  - struct CCurHint \* **pCCurHint**
  - int \* **aiCol**
  - struct [IdxCover](#) \* **pIdxCover**
- } **u**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.165 WalIterator::WalSegment Struct Reference

### Public Attributes

- int **iNext**
- ht\_slot \* **aIndex**
- u32 \* **aPgno**
- int **nEntry**
- int **iZero**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.166 WalWriter Struct Reference

Collaboration diagram for WalWriter:

### Public Attributes

- [Wal](#) \* **pWal**
- [sqlite3\\_file](#) \* **pFd**
- sqlite3\_int64 **iSyncPoint**
- int **syncFlags**
- int **szPage**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.167 WhereAndInfo Struct Reference

Collaboration diagram for WhereAndInfo:

### Public Attributes

- [WhereClause](#) **wc**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.168 WhereClause Struct Reference

Collaboration diagram for WhereClause:

### Public Attributes

- [WhereInfo](#) \* **pWInfo**
- [WhereClause](#) \* **pOuter**
- **u8 op**
- **int nTerm**
- **int nSlot**
- [WhereTerm](#) \* **a**
- [WhereTerm](#) **aStatic** [8]

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.169 WhereInfo Struct Reference

Collaboration diagram for WhereInfo:

## Public Attributes

- [Parse](#) \* **pParse**
- [SrcList](#) \* **pTabList**
- [ExprList](#) \* **pOrderBy**
- [ExprList](#) \* **pDistinctSet**
- **LogEst iLimit**
- **int aiCurOnePass [2]**
- **int iContinue**
- **int iBreak**
- **int savedNQueryLoop**
- **u16 wctrlFlags**
- **u8 nLevel**
- **i8 nOBSat**
- **u8 sorted**
- **u8 eOnePass**
- **u8 untestedTerms**
- **u8 eDistinct**
- **u8 bOrderedInnerLoop**
- **int iTop**
- [WhereLoop](#) \* **pLoops**
- **Bitmask revMask**
- **LogEst nRowOut**
- [WhereClause](#) **sWC**
- [WhereMaskSet](#) **sMaskSet**
- [WhereLevel](#) **a [1]**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.170 WhereLevel Struct Reference

Collaboration diagram for WhereLevel:

## Public Attributes

- **int iLeftJoin**
- **int iTabCur**
- **int iIdxCur**
- **int addrBrk**
- **int addrNxt**
- **int addrSkip**
- **int addrCont**
- **int addrFirst**
- **int addrBody**
- **u32 iLikeRepCntr**
- **int addrLikeRep**
- **u8 iFrom**
- **u8 op**
- **u8 p3**



- u8 **p5**
  - int **p1**
  - int **p2**
  - union {
    - struct {
      - int **nIn**
      - struct **InLoop** {
        - int **iCur**
        - int **addrInTop**
        - u8 **eEndLoopOp**
      - aInLoop**
    - in**
    - [Index](#) \* **pCovidx**
  - u**
- struct [WhereLoop](#) \* **pWLoop**
- Bitmask **notReady**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

## 3.171 WhereLoop Struct Reference

Collaboration diagram for WhereLoop:

### Public Attributes

- Bitmask **prereq**
  - Bitmask **maskSelf**
  - u8 **iTab**
  - u8 **iSortIdx**
  - LogEst **rSetup**
  - LogEst **rRun**
  - LogEst **nOut**
  - union {
    - struct {
      - u16 **nEq**
      - u16 **nBtm**
      - u16 **nTop**
      - [Index](#) \* **pIndex**
    - btree**
    - struct {
      - int **idxNum**
      - u8 **needFree**
      - i8 **isOrdered**
      - u16 **omitMask**
      - char \* **idxStr**
    - vtab**
  - u**
- u32 **wsFlags**

- u16 **nLTerm**
- u16 **nSkip**
- u16 **nLSlot**
- [WhereTerm](#) \*\* **aLTerm**
- [WhereLoop](#) \* **pNextLoop**
- [WhereTerm](#) \* **aLTermSpace** [3]

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.172 WhereLoopBuilder Struct Reference

Collaboration diagram for WhereLoopBuilder:

#### Public Attributes

- [WhereInfo](#) \* **pWInfo**
- [WhereClause](#) \* **pWC**
- [ExprList](#) \* **pOrderBy**
- [WhereLoop](#) \* **pNew**
- [WhereOrSet](#) \* **pOrSet**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.173 WhereMaskSet Struct Reference

#### Public Attributes

- int **n**
- int **ix** [BMS]

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.174 WhereOrCost Struct Reference

#### Public Attributes

- Bitmask **prereq**
- LogEst **rRun**
- LogEst **nOut**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.175 WhereOrInfo Struct Reference

Collaboration diagram for WhereOrInfo:

### Public Attributes

- [WhereClause](#) **wc**
- Bitmask **indexable**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.176 WhereOrSet Struct Reference

Collaboration diagram for WhereOrSet:

### Public Attributes

- u16 **n**
- [WhereOrCost](#) **a** [N\_OR\_COST]

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.177 WherePath Struct Reference

Collaboration diagram for WherePath:

### Public Attributes

- Bitmask **maskLoop**
- Bitmask **revLoop**
- LogEst **nRow**
- LogEst **rCost**
- LogEst **rUnsorted**
- i8 **isOrdered**
- [WhereLoop](#) \*\* **aLoop**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.178 WhereScan Struct Reference

Collaboration diagram for WhereScan:

#### Public Attributes

- [WhereClause](#) \* **pOrigWC**
- [WhereClause](#) \* **pWC**
- const char \* **zCollName**
- [Expr](#) \* **pIdxExpr**
- char **idxaff**
- unsigned char **nEquiv**
- unsigned char **iEquiv**
- u32 **opMask**
- int **k**
- int **aiCur** [11]
- i16 **aiColumn** [11]

The documentation for this struct was generated from the following file:

- `sqlite3.c`

### 3.179 WhereTerm Struct Reference

Collaboration diagram for WhereTerm:

#### Public Attributes

- [Expr](#) \* **pExpr**
- [WhereClause](#) \* **pWC**
- LogEst **truthProb**
- u16 **wtFlags**
- u16 **eOperator**
- u8 **nChild**
- u8 **eMatchOp**
- int **iParent**
- int **leftCursor**
- int **iField**
- union {
  - int **leftColumn**
  - [WhereOrInfo](#) \* **pOrInfo**
  - [WhereAndInfo](#) \* **pAndInfo**
 } **u**
- Bitmask **prereqRight**
- Bitmask **prereqAll**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.180 With Struct Reference

Collaboration diagram for With:

### Classes

- struct [Cte](#)

### Public Attributes

- int **nCte**
- [With](#) \* **pOuter**
- struct [With::Cte](#) **a** [1]

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.181 Parse::yColCache Struct Reference

### Public Attributes

- int **iTable**
- i16 **iColumn**
- u8 **tempReg**
- int **iLevel**
- int **iReg**
- int **lru**

The documentation for this struct was generated from the following file:

- `sqlite3.c`

## 3.182 YYMINORTYPE Union Reference

Collaboration diagram for YYMINORTYPE:

**Public Attributes**

- int **yyinit**
- sqlite3ParserTOKENTYPE **yy0**
- [Expr](#) \* **yy72**
- [TriggerStep](#) \* **yy145**
- [ExprList](#) \* **yy148**
- [SrcList](#) \* **yy185**
- [ExprSpan](#) **yy190**
- int **yy194**
- [Select](#) \* **yy243**
- [IdList](#) \* **yy254**
- [With](#) \* **yy285**
- struct [TrigEvent](#) **yy332**
- struct [LimitVal](#) **yy354**
- struct {
  - int **value**
  - int **mask**
 } **yy497**

The documentation for this union was generated from the following file:

- [sqlite3.c](#)

**3.183 yyParser Struct Reference**

Collaboration diagram for yyParser:

**Public Attributes**

- [yyStackEntry](#) \* **yytos**
- sqlite3ParserARG\_SDECL [yyStackEntry](#) **yystack** [YYSTACKDEPTH]

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

**3.184 yyStackEntry Struct Reference**

Collaboration diagram for yyStackEntry:

**Public Attributes**

- YYACTIONTYPE **stateno**
- YYCODETYPE **major**
- [YYMINORTYPE](#) **minor**

The documentation for this struct was generated from the following file:

- [sqlite3.c](#)

# Index

addPackage  
    Truck, [79](#)  
AgglInfo, [7](#)  
AgglInfo::AgglInfo\_col, [7](#)  
AgglInfo::AgglInfo\_func, [8](#)  
analysisInfo, [8](#)  
AuthContext, [8](#)  
AutoincInfo, [9](#)  
AuxData, [9](#)  
  
Bitvec, [9](#)  
BtCursor, [10](#)  
BtLock, [10](#)  
BtShared, [11](#)  
Btree, [11](#)  
BtreePayload, [11](#)  
BusyHandler, [12](#)  
  
CellArray, [12](#)  
CellInfo, [13](#)  
Client, [13](#)  
    Client, [15](#)  
    getAddress, [15](#)  
    getCoords, [15](#)  
    getReceivedPackages, [15](#)  
    getSentPackages, [15](#)  
    parseAddress, [16](#)  
    receivePackage, [16](#)  
    sendPackage, [16](#)  
    setAddress, [16](#)  
    toString, [16](#)  
CollSeq, [17](#)  
Column, [17](#)  
compareInfo, [18](#)  
CountCtx, [18](#)  
  
DateTime, [19](#)  
Db, [19](#)  
DbFixer, [19](#)  
deliverPackage  
    Truck, [79](#)  
DistinctCtx, [20](#)  
  
et\_info, [20](#)  
Expr, [20](#)  
ExprList, [21](#)  
ExprList::ExprList\_item, [22](#)  
ExprSpan, [22](#)  
  
FKey, [23](#)  
FKey::sColMap, [50](#)  
  
FileChunk, [22](#)  
FilePoint, [23](#)  
fts5\_api, [24](#)  
fts5\_tokenizer, [24](#)  
Fts5ExtensionApi, [24](#)  
Fts5Phraselater, [25](#)  
FuncDef, [25](#)  
FuncDefHash, [26](#)  
FuncDestructor, [26](#)  
  
Genetic, [26](#)  
getAddress  
    Client, [15](#)  
getCoords  
    Client, [15](#)  
getDirections  
    Truck, [80](#)  
getPackages  
    Truck, [80](#)  
getPriority  
    Package, [38](#)  
getPriorityString  
    Package, [38](#)  
getReceivedPackages  
    Client, [15](#)  
getReceiver  
    Package, [38](#)  
getSender  
    Package, [38](#)  
getSentPackages  
    Client, [15](#)  
getTruckID  
    Truck, [80](#)  
getWeight  
    Package, [38](#)  
    Truck, [80](#)  
  
Hash, [27](#)  
Hash::\_ht, [7](#)  
HashElem, [27](#)  
  
IdList, [27](#)  
IdList::IdList\_item, [28](#)  
IdxCover, [28](#)  
ImportCtx, [28](#)  
IncrMerger, [29](#)  
Incrblob, [29](#)  
Index, [29](#)  
IndexSample, [30](#)  
InitData, [30](#)

- IntegrityCk, [31](#)
- KeyInfo, [31](#)
- LimitVal, [32](#)
- Lookaside, [32](#)
- LookasideSlot, [32](#)
- Mem, [33](#)
- Mem::MemValue, [34](#)
- MemJournal, [33](#)
- MemPage, [34](#)
- MergeEngine, [35](#)
- Module, [35](#)
- NameContext, [35](#)
- PCache, [43](#)
- PCache1, [44](#)
- PGroup, [45](#)
- Package, [36](#)
  - getPriority, [38](#)
  - getPriorityString, [38](#)
  - getReceiver, [38](#)
  - getSender, [38](#)
  - getWeight, [38](#)
  - Package, [37](#)
  - setPriority, [39](#)
  - setReceiver, [39](#)
  - setSender, [39](#)
  - setWeight, [39](#)
- Pager, [40](#)
- PagerSavepoint, [41](#)
- Parse, [42](#)
- Parse::yColCache, [99](#)
- parseAddress
  - Client, [16](#)
- PgFreeslot, [44](#)
- PgHdr, [44](#)
- PgHdr1, [45](#)
- PmaReader, [46](#)
- PmaWriter, [46](#)
- PreUpdate, [47](#)
- PrintfArguments, [47](#)
- randomPackageEnum, [47](#)
- receivePackage
  - Client, [16](#)
- ReusableSpace, [48](#)
- RowSet, [48](#)
- RowSetChunk, [48](#)
- RowSetEntry, [49](#)
- SQLiteThread, [71](#)
- SavedModelInfo, [49](#)
- Savepoint, [49](#)
- ScanStatus, [50](#)
- Schema, [50](#)
- ScratchFreeslot, [51](#)
- Select, [51](#)
- SelectDest, [51](#)
- sendPackage
  - Client, [16](#)
- setAddress
  - Client, [16](#)
- setDirections
  - Truck, [80](#)
- setPriority
  - Package, [39](#)
- setReceiver
  - Package, [39](#)
- setSender
  - Package, [39](#)
- setWeight
  - Package, [39](#)
- ShellState, [52](#)
- SortCtx, [53](#)
- SortSubtask, [54](#)
- SorterFile, [53](#)
- SorterList, [54](#)
- SorterRecord, [54](#)
- sqlite3, [55](#)
- sqlite3::sqlite3InitInfo, [71](#)
- sqlite3\_api\_routines, [56](#)
- sqlite3\_backup, [61](#)
- sqlite3\_context, [61](#)
- sqlite3\_file, [62](#)
- sqlite3\_index\_info, [62](#)
- sqlite3\_index\_info::sqlite3\_index\_constraint, [62](#)
- sqlite3\_index\_info::sqlite3\_index\_constraint\_usage, [62](#)
- sqlite3\_index\_info::sqlite3\_index\_orderby, [63](#)
- sqlite3\_io\_methods, [64](#)
- sqlite3\_mem\_methods, [64](#)
- sqlite3\_module, [65](#)
- sqlite3\_mutex, [65](#)
- sqlite3\_mutex\_methods, [65](#)
- sqlite3\_pcache\_methods, [66](#)
- sqlite3\_pcache\_methods2, [66](#)
- sqlite3\_pcache\_page, [67](#)
- sqlite3\_rtree\_geometry, [67](#)
- sqlite3\_rtree\_query\_info, [68](#)
- sqlite3\_vfs, [68](#)
- sqlite3\_vtab, [69](#)
- sqlite3\_vtab\_cursor, [69](#)
- Sqlite3Config, [69](#)
- SrcCount, [71](#)
- SrcList, [72](#)
- SrcList::SrcList\_item, [72](#)
- Stat4Accum, [73](#)
- Stat4Sample, [73](#)
- StrAccum, [74](#)
- SubProgram, [74](#)
- SumCtx, [74](#)
- TabResult, [76](#)
- Table, [75](#)
- TableLock, [75](#)
- toString
  - Client, [16](#)



- Truck, [81](#)
- Token, [76](#)
- TrigEvent, [76](#)
- Trigger, [76](#)
- TriggerPrg, [77](#)
- TriggerStep, [77](#)
- Truck, [78](#)
  - addPackage, [79](#)
  - deliverPackage, [79](#)
  - getDirections, [80](#)
  - getPackages, [80](#)
  - getTruckID, [80](#)
  - getWeight, [80](#)
  - setDirections, [80](#)
  - toString, [81](#)
  - Truck, [79](#)
  - truckCount, [81](#)
- truckCount
  - Truck, [81](#)
- unixFile, [81](#)
- unixFileId, [82](#)
- unixInodeInfo, [82](#)
- unixShm, [82](#)
- unixShmNode, [83](#)
- UnixUnusedFd, [83](#)
- UnpackedRecord, [83](#)
- VTable, [89](#)
- ValueNewStat4Ctx, [84](#)
- Vdbe, [84](#)
- VdbeCursor, [86](#)
- VdbeFrame, [86](#)
- VdbeOp, [87](#)
- VdbeOp::p4union, [36](#)
- VdbeOpList, [88](#)
- VdbeSorter, [88](#)
- VtabCtx, [88](#)
- vxworksFileId, [89](#)
- Wal, [90](#)
- WalCkptInfo, [90](#)
- WalIndexHdr, [91](#)
- WalIterator, [91](#)
- WalIterator::WalSegment, [92](#)
- WalWriter, [92](#)
- Walker, [91](#)
- WhereAndInfo, [93](#)
- WhereClause, [93](#)
- WhereInfo, [93](#)
- WhereLevel, [94](#)
- WhereLoop, [95](#)
- WhereLoopBuilder, [96](#)
- WhereMaskSet, [96](#)
- WhereOrCost, [96](#)
- WhereOrInfo, [97](#)
- WhereOrSet, [97](#)
- WherePath, [97](#)
- WhereScan, [98](#)
- WhereTerm, [98](#)
- With, [99](#)
- With::Cte, [18](#)
- YYMINORTYPE, [99](#)
- yyParser, [100](#)
- yyStackEntry, [100](#)