

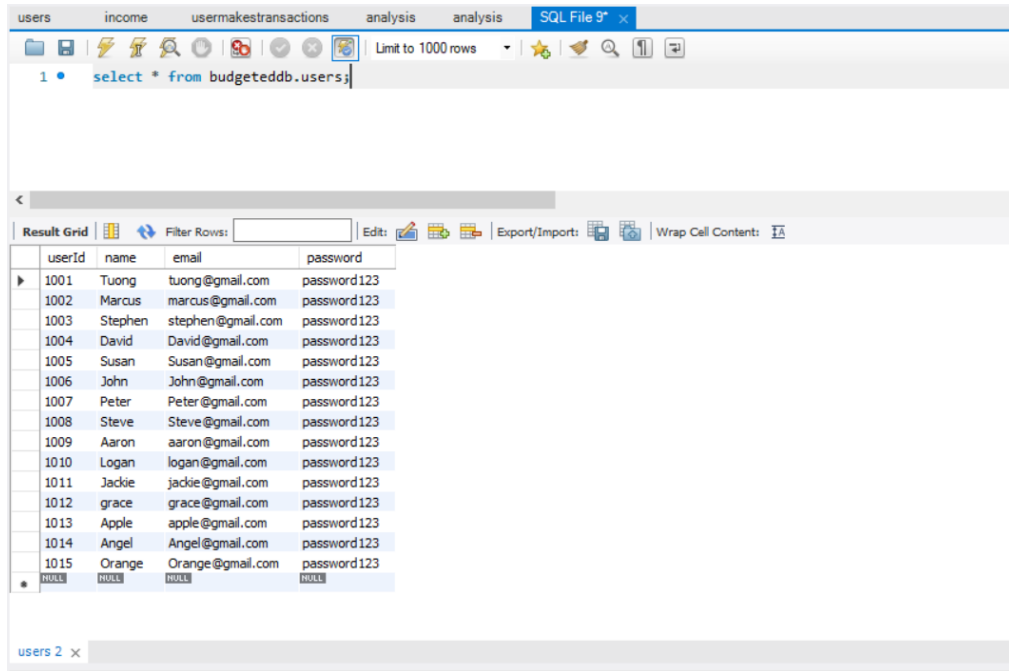
Tuong Nguyen

Marcus Zhou

Zhehua Liu

CS157A-Team-12

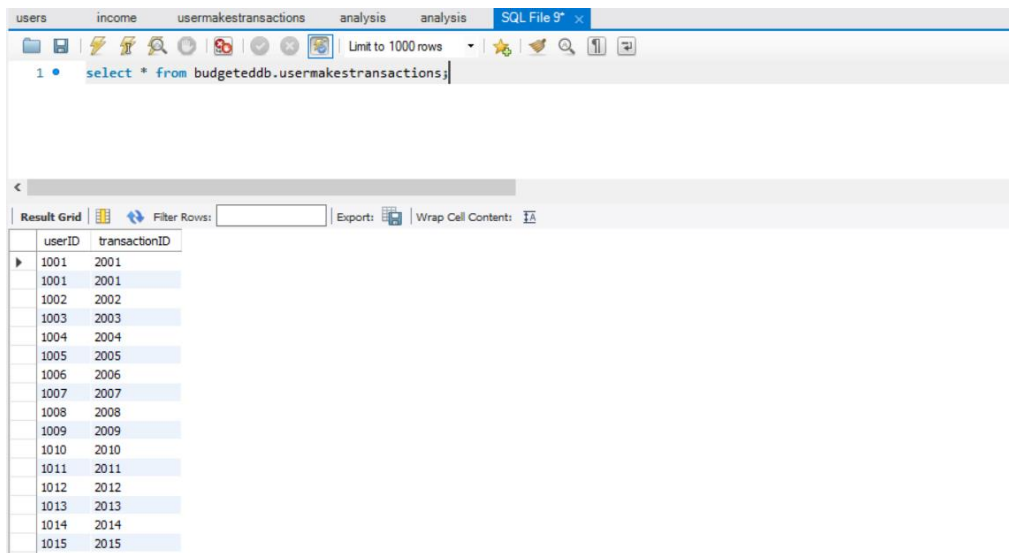
User



The screenshot shows a SQL IDE window with a query editor at the top containing the SQL statement: `select * from budgeteddb.users;`. Below the editor is a toolbar with icons for file operations, a 'Limit to 1000 rows' dropdown, and search/execution icons. The main area displays a 'Result Grid' with a table of user data. The table has four columns: `userId`, `name`, `email`, and `password`. It contains 15 rows of data, each with a unique `userId` and a common `password123`. A status bar at the bottom indicates 'users 2'.

userId	name	email	password
1001	Tuong	tuong@gmail.com	password123
1002	Marcus	marcus@gmail.com	password123
1003	Stephen	stephen@gmail.com	password123
1004	David	David@gmail.com	password123
1005	Susan	Susan@gmail.com	password123
1006	John	John@gmail.com	password123
1007	Peter	Peter@gmail.com	password123
1008	Steve	Steve@gmail.com	password123
1009	Aaron	aaron@gmail.com	password123
1010	Logan	logan@gmail.com	password123
1011	Jackie	jackie@gmail.com	password123
1012	grace	grace@gmail.com	password123
1013	Apple	apple@gmail.com	password123
1014	Angel	Angel@gmail.com	password123
1015	Orange	Orange@gmail.com	password123

Usermakestransactions



The screenshot shows a SQL IDE window with a query editor at the top containing the SQL statement: `select * from budgeteddb.usermakestransactions;`. Below the editor is a toolbar with icons for file operations, a 'Limit to 1000 rows' dropdown, and search/execution icons. The main area displays a 'Result Grid' with a table of transaction data. The table has two columns: `userID` and `transactionID`. It contains 15 rows of data, showing a sequence of transaction IDs (2001 to 2015) associated with the first user (1001). A status bar at the bottom indicates 'users 2'.

userID	transactionID
1001	2001
1001	2001
1002	2002
1003	2003
1004	2004
1005	2005
1006	2006
1007	2007
1008	2008
1009	2009
1010	2010
1011	2011
1012	2012
1013	2013
1014	2014
1015	2015

Transactions

users income usermakestransactions analysis analysis SQL File 9* x

Limit to 1000 rows

```
1 • budgeteddb.transactions;
```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content: [F1](#)

	transactionID	date	type	amount
▶	2001	10-10-2019	Income	30
	2002	10-10-2019	Expense	30
	2003	10-11-2019	Income	20
	2004	10-11-2019	Expenses	20
	2005	10-11-2019	Income	50
	2006	10-12-2019	Expenses	30
	2007	10-12-2019	Expenses	20
	2008	10-13-2019	Income	70
	2009	10-13-2019	Saving	70
	2010	10-14-2019	Expenses	30
	2011	10-14-2019	Expenses	20
	2012	10-15-2019	Income	50
	2013	10-16-2019	Income	50
	2014	10-17-2019	Expenses	30
	2015	10-18-2019	Income	60
*	NULL	NULL	NULL	NULL

Userhasbudget

users income usermakestransactions analysis analysis SQL File 9* x

Limit to 1000 rows

```
1 • select * from budgeteddb.userhasbudget;
```

Result Grid Filter Rows: Export: Wrap Cell Content: [F1](#)

	userID	BudgetID
▶	1001	3001
	1002	3002
	1003	3003
	1004	3004
	1005	3005
	1006	3006
	1007	3007
	1008	3008
	1009	3009
	1010	3010
	1011	3011
	1012	3012
	1013	3013
	1014	3014
	1015	3015

userhasbudget4 x

Budget

```
users  income  usermakestransactions  analysis  analysis  SQL File 9* x
1 • select * from budgeteddb.budget;
```

Result Grid			
Filter Rows:			
	budgetID	savingAcct	balance
▶	3001	0	30
	3002	0	0
	3003	0	20
	3004	0	0
	3005	0	50
	3006	0	20
	3007	0	0
	3008	0	70
	3009	70	70
	3010	70	40
	3011	70	20
	3012	70	70
	3013	70	120
	3014	70	90
	3015	70	150
*	NULL	NULL	NULL

Budgethasexpense_income

```
users  income  usermakestransactions  analysis  analysis  SQL File 9* x
1 • select * from budgeteddb.budgethasexpense_income;
```

Result Grid			
Filter Rows:			
	budgetID	incomeID	expenseID
▶	3001	4101	3201
	3002	4102	3202
	3003	4103	3203
	3004	4104	3204
	3005	4105	3205
	3006	4106	3206
	3007	4107	3207
	3008	4108	3208
	3009	4109	3209
	3010	4110	3210
	3011	4111	3211
	3012	4112	3212
	3013	4113	3213
	3014	4114	3214
	3015	4115	3215

Expenses

users income usermakestransactions analysis analysis SQL File 9* x

Limit to 1000 rows

```
1 • select * from budgeteddb.expenses;
```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell C

	expenseID	type	amount
▶	3201	netflix rental	30
	3202	breakfast	30
	3203	lunch	30
	3204	dinner	30
	3205	snack	30
	3206	water	30
	3207	breakfast	30
	3208	lunch	30
	3209	dinner	30
	3210	water	30
	3211	snack	30
	3212	breakfast	30
	3213	lunch	30
	3214	dinner	30
	3215	snack	30
*	NULL	NULL	NULL

Income

users income usermakestransactions analysis analysis SQL File 9* x

Limit to 1000 rows

```
1 • select * from budgeteddb.income;
```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content: IA

	IncomeID	Type	Amount
▶	4101	paycheck	30
	4102	paycheck	30
	4103	paycheck	30
	4104	paycheck	30
	4105	paycheck	30
	4106	paycheck	30
	4107	investment	30
	4108	paycheck	30
	4109	paycheck	30
	4110	paycheck	30
	4111	paycheck	30
	4112	paycheck	30
	4113	investment	30
	4114	investment	30
	4115	investment	30
*	NULL	NULL	NULL

Analysisusestrans_budget

users income usermakestransactions analysis analysis SQL File 9* x

Limit to 1000 rows

```
1 • select * from budgeteddb.analysisusestrans_budget;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

	analysisID	transactionID	budgetID
▶	5001	2001	3001
	5001	2001	3001
	5001	2001	3001
	5001	2001	3001
	5002	2002	3001
	5003	2003	3001
	5004	2004	3001
	5005	2005	3001
	5006	2006	3001
	5007	2007	3001
	5008	2008	3001
	5009	2009	3001
	5010	2010	3001
	5011	2011	3001
	5012	2012	3001
	5013	2013	3001
	5014	2014	3001
	5015	2015	3001

analysisusestrans_budget 10 x

Analysis

users income usermakestransactions analysis analysis SQL File 9* x

Limit to 1000 rows

```
1 • budgeteddb.analysis;
```

Result Grid Filter Rows: Edit: Export/Import: Wrap C

	analysisID	timeperiod	surplus	overdraft
▶	5001	Daily	30	0
	5002	Weekly	30	0
	5003	Monthly	50	0
	5004	Daily	0	20
	5005	Monthly	30	0
	5006	Weekly	0	50
	5007	Daily	30	0
	5008	Weekly	0	10
	5009	Monthly	30	0
	5010	Daily	30	0
	5011	Daily	0	40
	5012	Weekly	30	0
	5013	Monthly	0	50
	5014	Daily	0	20
	5015	Weekly	50	0
•	NULL	NULL	NULL	NULL

analysis 11 x