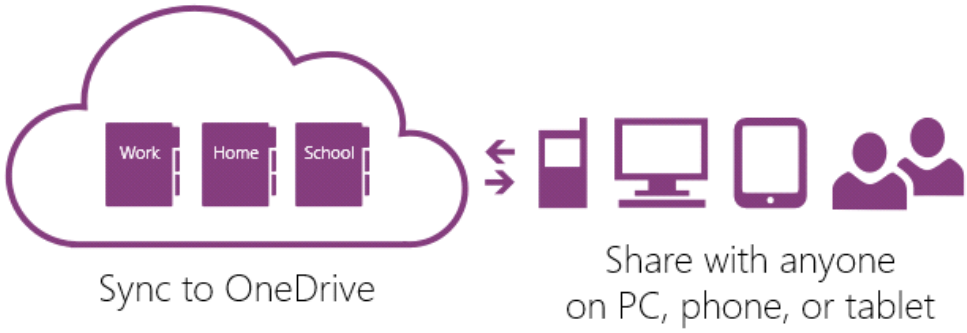
# OneNote: one place for all of your notes




Watch the
2 minute video

## 1. Take notes anywhere on the page

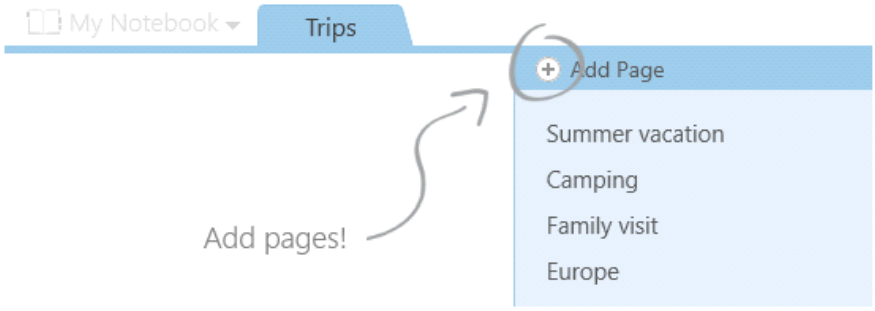
Write your name here

## 2. Get organized

You start with "My Notebook" - everything lives in here

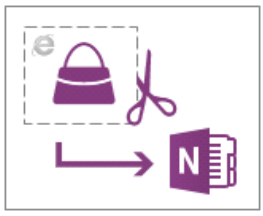

Add sections for activities like:


Add pages inside of each section:

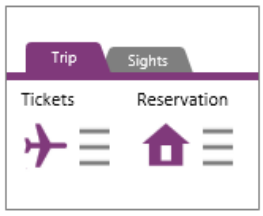



(Pages are over there)

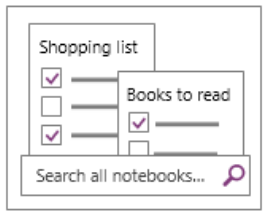
## 3. For more tips, check out 30 second videos



Clip from the web




Plan a trip with others



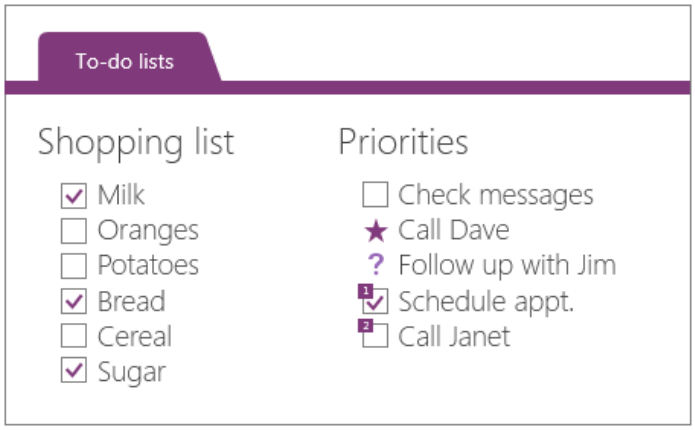

Search notes instantly


Write notes on slides

## 4. Create your first page

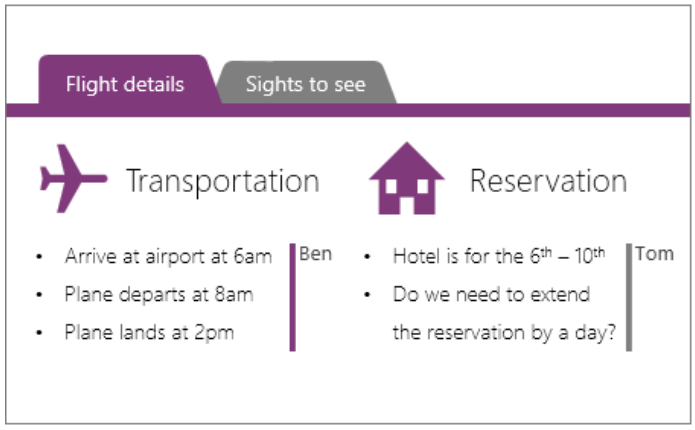
You're in the Quick Notes section - use it for random notes

# OneNote Basics



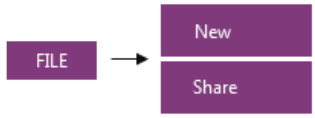

## Remember everything

- Add Tags to any notes
- Make checklists and to-do lists
- Create your own custom tags

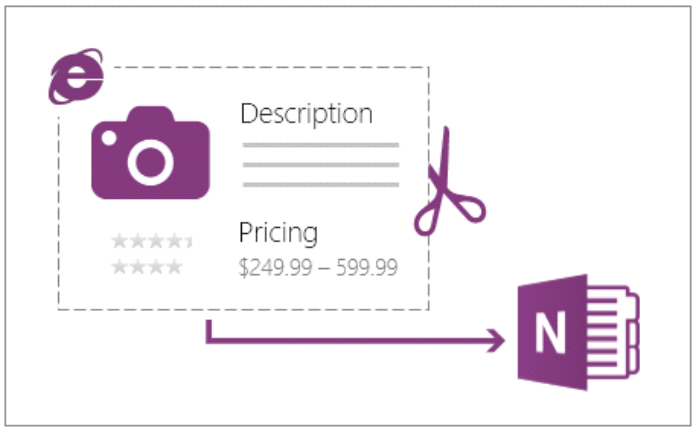

## Collaborate with others

- Keep your notebooks on OneDrive
- Share with friends and family
- Anyone can edit in a browser




## Keep everything in sync

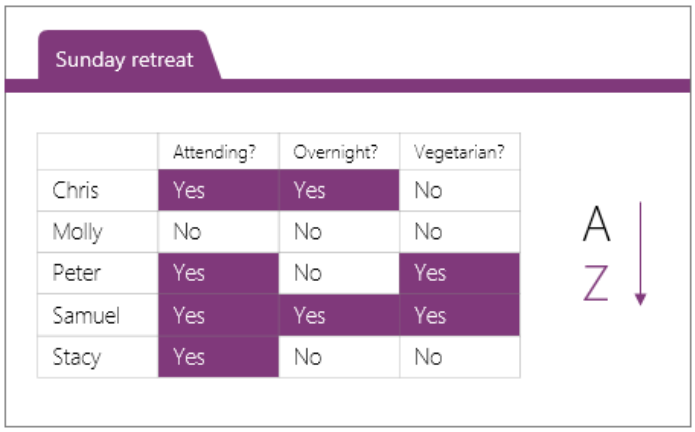
- People can edit pages at the same time
- Real-Time Sync on the same page
- Everything stored in the cloud
- Accessible from any device




## Clip from the web

- Quickly clip anything on your screen
- Take screenshots of products online
- Save important news articles

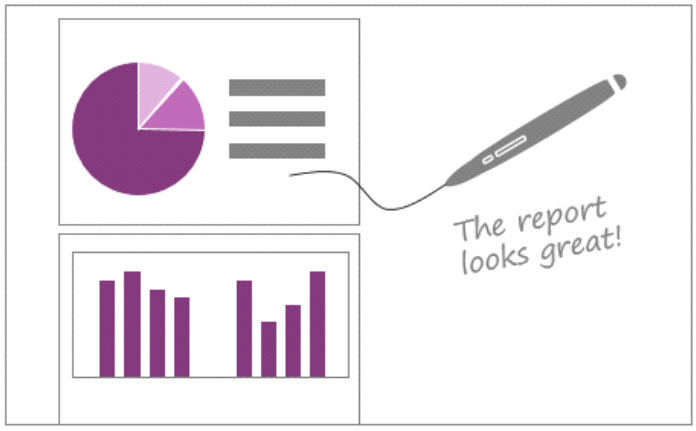
in your taskbar
OR
+ S on your keyboard




## Organize with tables

- Type, then press TAB to create a table
- Quickly sort and shade tables
- Convert tables to Excel spreadsheets

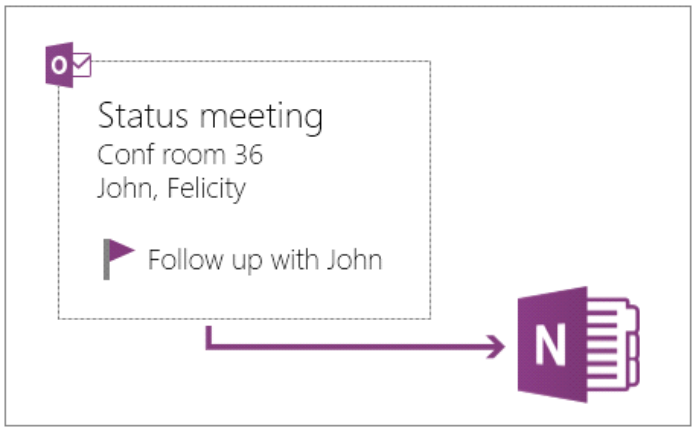
TABLE TOOLS
LAYOUT
Shading
Sort
Convert to Excel Spreadsheet




## Write notes on slides

- Send PowerPoint or Word docs to OneNote
- Annotate with a stylus on your tablet
- Highlight and finger-paint

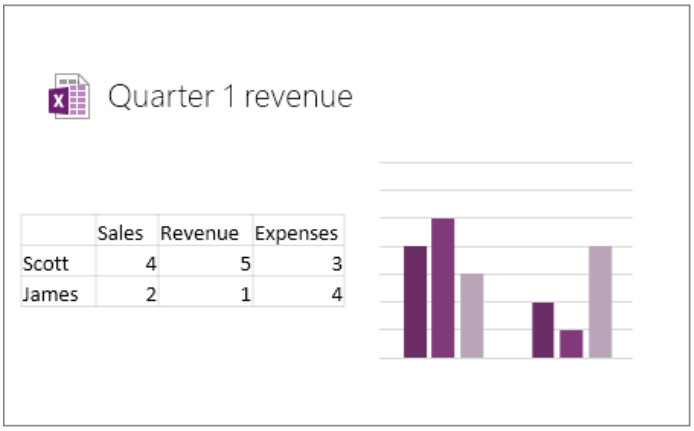
in your taskbar
OR
+ N on your keyboard




## Integrate with Outlook

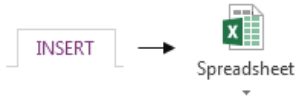
- Take notes on Outlook or Lync meetings
- Insert meeting details
- Add Outlook tasks from OneNote

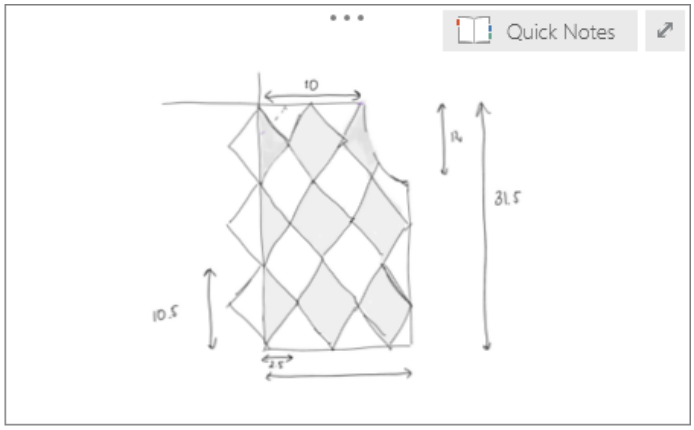

From Outlook:


## Add Excel spreadsheets

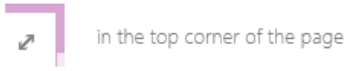
- Track finances, budgets, & more
- Preview updates on the page

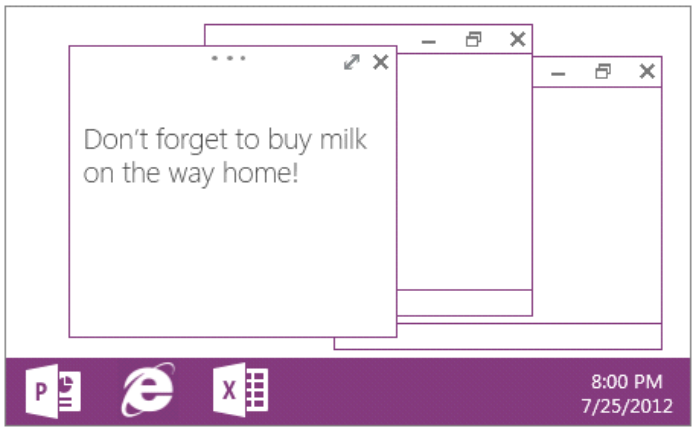



## Brainstorm without clutter

- Hide everything but the essentials
- Extra space to focus on your notes






# Take quick notes

- Quickly jot down thoughts and ideas
- They go into your Quick Notes section

in your taskbar

OR

+ N on your keyboard

Monday, April 8, 2019 4:41 PM

# Using workbench in MySQL 8 to dump database issues

**Description**:
Unknown table 'COLUMN_STATISTICS' in information_schema
**Solved**:
Using command from mysqldump. For example
**C:\Program Files\MySQL\MySQL Workbench 8.0 CE>mysqldump.exe --column-statistics=0 --host=35.241.86.208 --user=vinticket --password=XIPz4qqRHf --databases vinticket**

# Golang App

Monday, March 11, 2019 12:01 PM

Project structures
1) https://rakyll.org/style-packages/
2) https://medium.com/@benbjohnson/standard-package-layout-7cdbc8391fc1#.ds38va3pp
3) https://peter.bourgon.org/go-best-practices-2016/#repository-structure
4)https://www.ardanlabs.com/blog/2017/02/design-philosophy-on-packaging.html

Wednesday, July 17, 2019 11:31 AM

Commands

Databases

1) Ticket QC/UAT
   https://pma.vinid.dev/index.php
   vinticket_dev / 3lHJLTCtuh
   vinticket_qc / XIPz4qqRHf
   vinticket_uat/ unMCHutC61
   vinticket/ii5uHac7j3
2) Remote access via bastion on IP IP:172.31.0.16
   - gcloud compute ssh --zone=asia-east1-c --project=vinid-app-common-dev bastion
     - gcloud compute --project "vinid-app-common-dev" ssh --zone "asia-east1-c" "bastion" --ssh-flag="-L 8888:127.0.0.1:8888" --ssh-flag="-L 3308:172.31.0.16:3306"

     - gcloud compute --project "vinid-app-common-prod" ssh --zone "asia-east1-c" "bastion" --ssh-flag="-L 3308:172.31.1.3:3306" --tunnel-through-iap

Go commands

1) Verify golint
D:\company\projetcs\ticket\vinticket-services\src>golint ./…
2) run unit test
D:\company\projetcs\ticket\vinticket-services\src>go test --cover -p 1 -v -failfast -coverprofile=src.cov ./...

## Gcloud commands

- Switch to GCP project,eg: switch to **vinticket-nonprod** project

**gcloud config set project vinticket-nonprod**

- Pull credential from project to local

**gcloud container clusters get-credentials main --zone asia-east1-a --project <ten project> --internal-ip**

Eg: gcloud container clusters get-credentials main --zone asia-east1-a --project vinid-app-common-dev --internal-ip

- Export eviroment variable : export HTTPS_PROXY=http://localhost:8888

- Access GKE via bastion from project **vinid-app-common-dev(nonprod), vinid-app-common-prod(prod)**
**gcloud compute --project "vinid-app-common-dev" ssh --zone "asia-east1-c" "bastion" --ssh-flag="-L 8888:127.0.0.1:8888"**

Kubectl rolling updates

- Get deployment at one namspace
  **kubectl get deployment vinticket-services -n dev**
- Modify resource to do rollout update pods
  **kubectl set image deployment vinticket-services vinticket-services=asia.gcr.io/vinid-devops/$JOB_NAME:$IMAGE_TAG**
  **kubectl set resources deployment nginx --limits cpu=200m,memory=512Mi --requests cpu=100m,memory=256Mi**
- Rolling update status on deployment
  **kubectl rollout status deployment vinticket-services -n dev**
- Rolling update history on deployment
  **kubectl rollout history deployment vinticket-services-ops -n dev**
  **kubectl rollout history deployment vinticket-services-ops --revision 78 -n dev**

Kubectl exec a pod

- Kill a container name in a pod

  ```
  kubectl exec -it [POD_NAME] -c [CONTAINER_NAME] -- /bin/sh -c "kill 1"
  ```

  From <https://stackoverflow.com/questions/46123457/restart-container-within-pod>

Quick reboot the services

Kubectl scale to zero replicas to reboot the service

**kubectl scale deployment vinticket-services-ops --replicas=0 -n qc**

Kubectl scale replicas to more than zero to restart the service

**kubectl scale deployment vinticket-services-ops --replicas=2 -n qc**

Tuesday, July 23, 2019 9:03 AM

## Locking

- InnoDB uses row level locking, the row level locking is **index record** lock that encounter s during query searching for records
- Two type of locking on row level(**actually index lock**) : **share lock** and **exclusive lock**
  - Some lock modes in MySQL 8.0:

"X" if the lock is on both record and gap (a.k.a. "Next Key Lock" in our documentation)
- "X,REC_NOT_GAP" if the lock is on record only (a.k.a. "Record Lock")
- "X,GAP" if the lock is on the gap only (a.k.a. "GAP Lock")
- "X,GAP,INSERT_INTENTION" if the lock is an insert intention lock

There are still some complications when the lock is on supremum, in which case we usually don't add ",GAP" suffix, even though the lock behaves like a ",GAP" lock.

From <https://bugs.mysql.com/bug.php?id=96013>

**Summary to discover:**

According to mysql reference https://dev.mysql.com/doc/refman/8.0/en/innodb-locks-set.html

1. For locking reads (SELECT with FOR UPDATE or FOR SHARE), UPDATE, and DELETE statements, for non -unique indexes, InnoDB locks the index range scanned, using gap locks or next -key locks to block insertions by other sessions into the gaps covered by the range.
2. UPDATE ... WHERE ... sets an exclusive next-key lock on every record the search encounters. However, only an index record lock is required for statements that lock rows using a unique index to search for a unique row.
3. DELETE FROM ... WHERE ... sets an exclusive next-key lock on every record the search encounters. However, only an index record lock is required for statements that lock rows using a unique index to search for a unique row

my question is: 1. when only gap lock will be used while when next key lock will be used and why? 2. what is the interaction process between mysql server layer and innodb lock ?
on same page:https://dev.mysql.com/doc/refman/8.0/en/innodb-locks-set.html

1. INSERT sets an exclusive lock on the inserted row. This lock is an index -record lock, not a next-key lock (that is, there is no gap lock) and does not prevent other sessions from inserting into the gap before the inserted row.
2. INSERT ... ON DUPLICATE KEY UPDATE differs from a simple INSERT in that an exclusive lock rather than a shared lock is placed on the row to be updated when a duplicate-key error occurs. An exclusive index-record lock is taken for a duplicate primary key value. An exclusive next-key lock is taken for a duplicate unique key value.

why is next-key lock used for a duplicate unique key when use INSERT ... ON DUPLICATE KEY UPDATE rather than record lock?

From <https://stackoverflow.com/questions/55426476/when-only-gap-lock-will-be-used-while-when-next-key-lock-will-be-used-and-why>

- Locking mechanisms :
  - Select on the field in which is primary key

    **Lock on write queries:**

    Lock on the existed value of primary key => lock mode: **X, RECORD_NOT_GAP**

    The primary key **id = 6059** matched

    select * from seats where id = 6059 for update;

| LOCK_TYPE | LOCK_MODE | LOCK_STATUS | LOCK_DATA |
|---|---|---|---|
| TABLE | IX | GRANTED | NULL |
| RECORD | X,REC_NOT_GAP | GRANTED | 6059 |

    delete from seats where id =6059 ;

| INDEX_NAME | OBJECT_INSTANCE_BEGIN | LOCK_TYPE | LOCK_MODE | LOCK_STATUS | LOCK_DATA |
|---|---|---|---|---|---|
| NULL | 2026904250264 | TABLE | IX | GRANTED | NULL |
| PRIMARY | 2026924419192 | RECORD | X,REC_NOT_GAP | GRANTED | 6059 |

    update seats set price=100000 where id = 6059;

| INDEX_NAME | OBJECT_INSTANCE_BEGIN | LOCK_TYPE | LOCK_MODE | LOCK_STATUS | LOCK_DATA |
|---|---|---|---|---|---|
| NULL | 2026904250264 | TABLE | IX | GRANTED | NULL |
| PRIMARY | 2026924419192 | RECORD | X,REC_NOT_GAP | GRANTED | 6059 |

    Lock on the non-existed value of primary key => lock mode: **X, GAP**
    This gap lock is between index records, or a lock on the gap before the first or after the last index record,
    the purpose of this gap lock is to prevent other transactions from inserting a value between the gap.

    The primary key **id =** 6057 is non-existed value and it will lock rows in the gap with id from **6054-> 6059**

    select * from seats where id = 6057 for update;

| INDEX_NAME | OBJECT_INSTANCE_BEGIN | LOCK_TYPE | LOCK_MODE | LOCK_STATUS | LOCK_DATA |
|---|---|---|---|---|---|
| NULL | 2026904250264 | TABLE | IX | GRANTED | NULL |
| PRIMARY | 2026924419192 | RECORD | X,GAP | GRANTED | 6059 |

    **Lock on read queries:**

    The primary key matched with the value => lock mode: **S,RECORD_NOT_GAP**

    select * from seats where id =6059 for share;

| INDEX_NAME | OBJECT_INSTANCE_BEGIN | LOCK_TYPE | LOCK_MODE | LOCK_STATUS | LOCK_DATA |
|---|---|---|---|---|---|
| NULL | 2026904250264 | TABLE | IS | GRANTED | NULL |
| PRIMARY | 2026924419192 | RECORD | S,REC_NOT_GAP | GRANTED | 6059 |

    The primary key did not match with the value => lock mode: **S, GAP**
    This gap lock is between index records, or a lock on the gap before the first or after the last index record, the purpose of this gap lock is to prevent other transactions from inserting a value between the gap.

    The primary key **id =** 6057 is non-existed value and it will lock rows in the gap with id from **6054-> 6059**

    select * from seats where id =6057 for share;

| INDEX_NAME | OBJECT_INSTANCE_BEGIN | LOCK_TYPE | LOCK_MODE | LOCK_STATUS | LOCK_DATA |
|---|---|---|---|---|---|
| NULL | 2026904250264 | TABLE | IS | GRANTED | NULL |
| PRIMARY | 2026924419192 | RECORD | S,GAP | GRANTED | 6059 |

  - Select on the fields in which are unique key
  - Select on the fields in which are indexed fields

## Indexing

- MyISAM indexes refer to the indexed rows by their physical storage locations, but InnoDB refers to them by their primary key values

## Transaction

## Techniques

## Performance

Defer Join:

**Explain SQL**

Thursday, August 1, 2019 11:06 AM

https://medium.com/@devfire/how-to-become-a-devops-engineer-in-six-months-or-less-366097df7737

Friday, December 13, 2019 2:44 PM

Worker

Use command to install wrangler

```
sudo npm i --save [packagename] --unsafe-perm=true --allow-root
```